

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Monitoring Systems for Parallel Distributed Data Management Systems

Emanuel Augusto Santos Pinho



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Alexandre Miguel Barbosa Valle de Carvalho

29 de Julho de 2016

Monitoring Systems for Parallel Distributed Data Management Systems

Emanuel Augusto Santos Pinho

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: António Coelho

Arguente: Luís Magalhães

29 de Julho de 2016

Resumo

Um sistema de informação *Big Data* é geralmente composto por um conjunto alargado de máquinas que, funcionando em rede, desempenham funções distribuídas e em paralelo. Estes sistemas estão vulgarmente acompanhados de sistemas de monitorização para avaliação de *performance* e prevenção de erros.

Na atualidade existem inúmeros sistemas que desempenham funções de monitorização, no entanto, apresentam algumas desvantagens como limitações ao nível da representação visual dos dados monitorizados, organização visual complexa dos dados e a sua abordagem direcionada a componentes físicos.

O presente estudo tem como principal objetivo avaliar alternativas de mecanismos visuais gráficos e de interação que permitam a representação de dados de monitorização de *performance* em ambientes de *grid computing*, permitindo fornecer ao utilizador final informação capaz de contribuir de forma objetiva na perceção da análise comportamental do sistema.

O estudo insere-se no contexto europeu de um projeto FP7 de nome LeanBigData (contando já com duas classificações anuais de excelente) que tem como principal objetivo a construção de um sistema de informação *Big Data* completo, focando a atenção na resolução de problemas de eficiência. Acompanhado deste sistema de informação pretende-se um sistema de monitorização capaz de dar resposta aos objetivos e necessidades do projeto, que se apresentam como diferenciadoras num contexto de resolução de problemas.

Começando pela análise de algumas das soluções mais usadas, incluindo detalhes sobre as qualidades e defeitos, este documento apresenta também a esquematização de uma prova de conceito que responde aos problemas de visualização encontrados nas ferramentas. Foram estudadas as melhores alternativas para conseguir manter a interação do sistema com o utilizador simples mas ao mesmo tempo eficaz, melhorando as alternativas de representação geral do sistema, onde o utilizador além de perceber quais os componentes que integram o sistema, consegue relacioná-los de forma a perceber como melhorar o seu desempenho geral. Foi idealizada uma alternativa à representação da organização física dos diversos componentes do sistema, reproduzindo uma interpretação lógica onde os recursos estão associados a competências/responsabilidades de componentes no sistema monitorado. Para que a interação se concluísse simples mas completa, a construção e edição de diversos gráficos e respetivos *dashboards* também foi requerida como abrangente nas suas potencialidades de diferenciação e manipulação.

Com a implementação da prova de conceito proposta e respetiva aprovação pelo consórcio do projeto, foram realizados os devidos testes de simulação comportamental em um sistema de teste, simulando o objeto de estudo. A proposta apresentada resolve as problemáticas identificadas, facilitando o uso de uma ferramenta de monitorização. A implementação do sistema é acompanhada pela participação em uma conferência científica e pelo desenvolvimento de uma ferramenta *open source* de apoio à criação gráfica em ambientes *web*.

Abstract

A Big Data information system is usually comprised by a large number of machines, which closely in a network, perform distributed and parallel operations. Usually these systems go together with a monitor system to evaluate performance and prevent errors. Nowadays, there are many systems which perform these functions, however, they have some disadvantages such as the graphics limitation, complex organization and its targeted approach to physical components.

This study aims to understand what are the graphic visual mechanisms and interaction processes that allow the representation of performance monitoring data in grid computing environments, allowing the end-user information the ability to easily perceive the behavior of the Big Data information system.

This study is being carried in the broader context of a European project FP7 LeanBigData (which has already been graded with two excellents, in annual revision classifications) that aims to build a complete BigData information system, focusing its attention on solving efficiency problems. The monitor system innovative solution must be able to deal with the project needs, which are presented as a distinctive in problem-solving context.

This study begins with an analysis of some of the systems used today. Based on their main qualities and disadvantages, it was formulated a hypothesis, conceived a solution and developed a proof of concept, where they were applied different techniques of representation of the resources structure, to allow the user to understand not only the system's organization, but also the relation between the components. It was also devised an alternative approach to the representation of the physical components, designing a logical representation, which can be easily drawn conclusions on task organization of the network machines. To keep this interaction simple and complete, it requires a construction and edition of graphs and dashboards to enhance differentiation and manipulation capacities.

With the implementation of the proof of concept proposed and approved by the project consortium, appropriate simulation tests were made with a simulation system carried out by the project. The proposal solve the problems identified and brings up a better approach in terms of monitoring tools. The development of the system was followed by a participation in a scientific conference and with the development of a new open source tool to chart development in web environments.

Agradecimentos

É o meu nome que vem destacado na autoria deste trabalho, no entanto, o seu desenvolvimento não era possível sem o apoio prestado por algumas pessoas que fizeram parte do meu percurso académico e da minha vida. Sem elas, a realização desta dissertação seria uma tarefa impossível.

Em primeiro lugar quero agradecer à minha família pela presença e apoio que revelaram, não só ao longo da realização desta dissertação, mas ao longo de todo o meu percurso enquanto estudante e pessoa. Eles foram, sem dúvida, o principal pilar de sustento inerente ao peso desta importante fase. O meu agradecimento acompanha o pedido de que assim seja ao longo de muitas mais etapas que se avizinham.

Em segundo, ao meu orientador professor Alexandre Carvalho, agradeço todas as críticas, sugestões e total disponibilidade. A sua exigência e experiência na área da investigação foram aspetos essenciais para que o desenvolvimento do trabalho fosse cumprido da melhor forma.

Sem os meus amigos, colegas de curso e parceiros de investigação no INESC este trabalho seria muito mais difícil. A eles, um obrigado pelas sugestões, apoios e momentos de lazer proporcionados. Também eles contribuíram para que o desenrolar desta etapa fosse encarado, todos os dias, com o mesmo otimismo.

Por fim, agradeço a todas as pessoas que não descrevo pessoalmente, mas que estiveram presentes e contribuíram para a conclusão desta importante etapa.

Esta caminhada tem um passo de cada um de vós.

Emanuel Pinho

*“I may not be there yet,
but I’m closer than I was yesterday.”*

José N. Harris

Conteúdo

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 1.1 | Contexto e Motivação | 1 |
| 1.2 | O Projeto LeanBigData | 2 |
| 1.3 | Problema e objetivos | 3 |
| 1.4 | Estrutura da Dissertação | 4 |
| 2 | Revisão Bibliográfica | 5 |
| 2.1 | Técnicas de visualização | 5 |
| 2.2 | Sistemas de Monitorização | 8 |
| 2.2.1 | Sistemas de monitorização | 9 |
| 2.2.2 | Ferramentas de visualização | 14 |
| 2.3 | Tecnologias de visualização | 16 |
| 2.3.1 | D3.js | 16 |
| 2.3.2 | Chartlist.js | 17 |
| 2.3.3 | Google Charts | 17 |
| 2.3.4 | Gridster | 17 |
| 2.4 | Sumário | 17 |
| 3 | Descrição do problema | 19 |
| 3.1 | Limitações dos sistemas de monitorização | 19 |
| 3.2 | Proposta de solução | 20 |
| 3.3 | Sumário | 22 |
| 4 | Especificação da solução proposta | 23 |
| 4.1 | Estudo da interface | 23 |
| 4.2 | Diagrama de fluxo | 25 |
| 4.3 | Representação visual do sistema monitorado | 25 |
| 4.3.1 | Representação hierárquica | 26 |
| 4.3.2 | Representação relacional | 27 |
| 4.3.3 | Representação dimensional | 29 |
| 4.3.4 | Representação por listas | 29 |
| 4.3.5 | Representação lógica | 30 |
| 4.4 | Filtragem de componentes | 33 |
| 4.5 | Janela de componentes selecionados | 33 |
| 4.6 | Tipos de gráfico de monitorização | 34 |
| 4.6.1 | Gráfico de linhas | 35 |
| 4.6.2 | Gráfico de linhas acumulativo | 36 |
| 4.6.3 | Gráfico de área | 36 |

CONTEÚDO

| | | |
|----------|---|-----------|
| 4.6.4 | Gráfico de barras | 37 |
| 4.6.5 | Gráfico de linhas e barras | 38 |
| 4.6.6 | Gráfico de barras horizontais | 38 |
| 4.6.7 | Gráfico circular | 39 |
| 4.6.8 | Personalização gráfica | 39 |
| 4.7 | <i>Dashboards</i> | 41 |
| 4.8 | Sumário | 42 |
| 5 | Implementação da solução | 45 |
| 5.1 | Avaliação do modo de comunicação Gangleia | 45 |
| 5.1.1 | Comunicação baseada em <i>socket</i> | 46 |
| 5.1.2 | Comunicação suportada por RRDTOOL | 46 |
| 5.1.3 | Comunicação suportada em Graphite | 47 |
| 5.1.4 | Sumário | 47 |
| 5.2 | Arquitetura | 47 |
| 5.3 | Modelos de dados | 48 |
| 5.4 | Pilha tecnológica | 49 |
| 5.5 | Desenvolvimento gráfico | 51 |
| 5.5.1 | Angular d3-charts | 51 |
| 5.6 | Desenvolvimento de <i>dashboards</i> | 51 |
| 5.7 | Sumário | 52 |
| 6 | Resultados | 53 |
| 6.1 | Objetivos alcançados | 54 |
| 6.2 | Artigos científicos | 55 |
| 7 | Conclusões | 57 |
| 7.1 | Trabalho Futuro | 58 |
| | Referências | 59 |
| A | Casos de uso | 63 |
| B | Artigo CISTI'2016 | 67 |

Lista de Figuras

| | | |
|------|--|----|
| 2.1 | Triângulo de visualização | 6 |
| 2.2 | Avaliação da estrutura centrada no utilizador | 7 |
| 2.3 | Disponibilidade cerebral na análise dos dados | 8 |
| 2.4 | Pormenor da interface Ganglia | 10 |
| 2.5 | Pormenor da interface Observium | 11 |
| 2.6 | Pormenor da interface VisPerf | 13 |
| 2.7 | Fluxo de dados no Graphite | 15 |
| 2.8 | Pormenor da interface Graphite | 15 |
| 2.9 | Pormenor da interface Grafana | 16 |
| 4.1 | Desenho do aspeto visual da solução | 24 |
| 4.2 | Diagrama do normal fluxo de uso | 25 |
| 4.3 | Vista de representação hierárquica | 26 |
| 4.4 | Vista de representação relacional | 27 |
| 4.5 | Vista de representação dimensional | 29 |
| 4.6 | Vista de representação por listas | 30 |
| 4.7 | Vista de representação lógica | 31 |
| 4.8 | Janela de componentes selecionados para a vista lógica | 32 |
| 4.9 | Janela de componentes selecionados | 33 |
| 4.10 | Lista de gráficos disponíveis | 34 |
| 4.11 | Exemplo de um cabeçalho de um gráfico | 35 |
| 4.12 | Gráficos de linhas | 35 |
| 4.13 | Gráfico de linhas acumulativo | 36 |
| 4.14 | Gráficos de área | 37 |
| 4.15 | Gráficos de barras | 38 |
| 4.16 | Gráfico de linhas e barras | 38 |
| 4.17 | Gráfico de barras horizontais | 39 |
| 4.18 | Gráfico circular | 39 |
| 4.19 | Menu de opções de um gráfico | 40 |
| 4.20 | Gráfico criado através da vista lógica | 41 |
| 4.21 | Vista exemplar de um Dashboard | 41 |
| 4.22 | Menu de opções dos <i>dashboards</i> | 42 |
| 5.1 | Diagrama de representação da arquitetura de testes | 46 |
| 5.2 | Diagrama de arquitetura do sistema desenvolvido | 48 |
| 5.3 | Modelo de dados do sistema desenvolvido | 49 |
| 5.4 | Arquitetura de comunicação do módulo dos <i>dashboards</i> | 52 |

LISTA DE FIGURAS

Lista de Tabelas

| | | |
|------|---|----|
| 2.1 | Resumo das características das ferramentas analisadas | 14 |
| A.1 | Caso de uso 1.1 | 63 |
| A.2 | Caso de uso 1.2 | 63 |
| A.3 | Caso de uso 1.3 | 63 |
| A.4 | Caso de uso 2.1 | 64 |
| A.5 | Caso de uso 2.2 | 64 |
| A.6 | Caso de uso 2.3 | 64 |
| A.7 | Caso de uso 2.4 | 64 |
| A.8 | Caso de uso 2.5 | 65 |
| A.9 | Caso de uso 2.6 | 65 |
| A.10 | Caso de uso 2.7 | 65 |
| A.11 | Caso de uso 2.8 | 65 |
| A.12 | Caso de uso 2.9 | 66 |
| A.13 | Caso de uso 2.10 | 66 |

LISTA DE TABELAS

Abreviaturas e Símbolos

| | |
|--------------------------|--|
| API | Application Programming Interface |
| CPU | Central Processing Unit |
| CSS | Cascading Style Sheets |
| DNS | Domain Name Server |
| DOM | Document Object Model |
| FTP | File Transfer Protocol |
| gmetad | Ganglia Meta Daemon |
| gmond | Ganglia Monitoring Daemon |
| GUI | Graphical User Interface |
| HTML | HyperText Markup Language |
| HTTP | Hypertext Transfer Protocol |
| I/O | Input/Output |
| INESC TEC | Institute of Engineering Systems and Computers, Technology and Science |
| JMX | Java Management Extensions |
| KPI | Key Performance Indicator |
| LBD | Lean Big Data |
| LeanXscale | Produto Lean Big Data |
| MDS | Monitoring & Discovery System |
| Métrica de monitorização | Valor de desempenho recolhido em intervalos de tempo regulares |
| NOSQL | Not Only SQL |
| OS | Operating System |
| PNG | Portable Network Graphics |
| POP3 | Post Office Protocol |
| RRD | Round Robin Database |
| SMTP | Simple Mail Transfer Protocol |
| SNMP | Simple Network Management Protocol |
| SQL | Structured Query Language |
| SSH | Secure Shell |
| SVG | Scalable Vector Graphics |
| TCP | Transmission Control Protocol |
| TI | Tecnologias de informação |
| XDR | eXternal Data Representation |

Capítulo 1

Introdução

Na atualidade, o termo *Big Data* é frequentemente usado no contexto das TI. Não se caracteriza unicamente por um sistema contendo um enorme volume de dados, mas também pela necessidade de funcionamento em um sistema eficiente e escalável, capaz de sustentar a estrutura requerida por estes sistemas. Mas na verdade, qual o motivo do seu uso? O mundo de hoje faz com que a quantidade de informação gerada eletronicamente aumente de dia para dia. Um exemplo desta dimensão são as redes sociais Instagram e Youtube, que, de acordo com Josh James em [Jam14], a cada minuto armazenam em média 216000 fotografias ou 72 horas de vídeo, respetivamente. A posse e tratamento destes dados, revelam-se um dos maiores valores competitivos do mercado. De acordo com Mark Zuckerberg, em [Zuc15]:

Second, we're working on AI because we think more intelligent services will be much more useful for you to use. For example, if we had computers that could understand the meaning of the posts in News Feed and show you more things you are interested in, that would be pretty amazing. Similarly, if we could build computers that could understand what's in an image and could tell a blind person who otherwise couldn't see that image that would be amazing as well. This is all within our reach and I hope we can deliver it in the next 10 years.

Para que este valor seja alcançado é necessário um exaustivo tratamento de dados: mecanismos de recolha de dados das redes sociais, onde os seus utilizadores expõe detalhes da vida, são acompanhados de sistemas de enorme capacidade de interpolação de dados [TSA⁺10].

1.1 Contexto e Motivação

No núcleo de um sistema de *Big Data* está presente um número elevado de repositórios, onde a informação é processada, armazenada e representada. Esses repositórios atuam em unidades distribuídas, chamados de *clusters*, que possibilitam o processamento da informação com elevado

desempenho, distribuindo as várias tarefas associadas ao processamento da informação pelos vários recursos existentes na rede. Estes sistemas fazem-se beneficiar de técnicas de *sharding*, caracterizadas pela divisão de um grande sistema de bases de dados, em vários de menor dimensão, produzindo melhor desempenho pelo efeito que resulta da paralelização aquando da execução dos processos. Com isto, a monitorização do estado de um sistema tão complexo é tão importante quanto as tarefas que estes realizam, tal como defende a página oficial do MongoDB, em [Monb]: *"Monitoring is a critical component of all database administration."*

A inevitabilidade da existência de um sistema de monitorização é unânime e referida por Chad Maynard em [CM] e Robert Williams em [RW] respetivamente: *"For Managed IT companies like ours, monitoring our customers' infrastructure is critical. For years, we went through every network-monitoring package we could find. We spent a lot of time and money, but we were never really happy with the results we got "e " a key part of our business is the delivery of high-capacity 'always-on' networking across all our locations. With a wide selection of hardware platforms and a high overall port count it is essential that we collect high value statistical information in an automated manner."*

O principal objetivo de um sistema de monitorização é fornecer dados sobre o estado do sistema monitorizado para que a perceção e prevenção de comportamentos e de anomalias seja muito mais simples, por forma a evitar estragos. Estes dados esperam-se representados com um design visual simples e objetivo, onde facilmente podem ser retiradas conclusões acerca do normal comportamento de determinado componente em análise, comparando-o com os demais sistemas envolventes, históricos guardados e ainda obter informação acerca da origem de eventuais alterações não expectáveis. É com essa finalidade que surge o WP3 do projeto LeanBigData [Lea14].

1.2 O Projeto LeanBigData

O projeto europeu FP7 LeanBigData (LBD) tem como objetivo principal o estudo e desenvolvimento de um sistema de *Big Data* ultra escalável e ultra eficiente. Este projeto é realizado pelo seguinte consórcio: Universidad Politecnica de Madrid Intel, Computer Associates (CA), Foundation for Research and Technology – Hellas, Institute of Engineering Systems and Computers (INESC TEC), SyncLab, Atos Spain S.A., Institute of Communication and Computer Systems e Portugal Telecom (PT).

Sendo um dos seus principais objetivos resolver problemas de eficiência comuns noutros sistemas de *Big Data* existentes na atualidade, o projeto apresenta-se de forma a:

- Melhorar o custo computacional inserido no processamento analítico para análise de sequências dinâmicas;
- Melhorar ferramentas de manipulação de Big data com elevados tempos de resposta;
- Fornecer uma solução end-to-end.

Para responder a estas problemáticas de forma eficiente, o consórcio do projeto propõe um novo sistema de base de dados *NoSQL* transacional, um sistema de processamento complexo de eventos e um motor distribuído de interrogações à base de dados.

Paralelamente a estas necessidades, foi também considerado importante o desenvolvimento de um sistema de monitorização para análise de desempenho, capaz de recolher, armazenar e disponibilizar informação acerca do estado do sistema, nomeadamente o tempo de execução e resposta, consumo de recursos, acessos e estados de componentes físicos e lógicos.

Com o aumento da complexidade de um sistema, a monitorização deve centrar-se em casos isolados mas também no sistema como um todo, possibilitando ainda uma análise da fonte do erro (*root-cause analysis*).

Embora existam alguns produtos de mercado que albergam estas situações, são ainda necessárias ferramentas capazes de fornecer a representação física mas também a lógica do sistema, com a possibilidade de cruzar informação e encontrar várias soluções possíveis para o melhor desempenho do sistema. A deteção da origem do problema pode ser uma tarefa complicada quando o sistema fornece uma complexidade visual grande. Por isso é necessário o desenvolvimento de uma solução baseada nas tecnologias emergentes na área da interação pessoa-computador, visualização, que torne a análise mais simples, intuitiva e eficiente.

1.3 Problema e objetivos

Na atualidade, quando se aborda ferramentas de monitorização, na verdade, dificilmente se encontram aspetos verdadeiramente inovadores. Então, qual é a verdadeira razão da existência deste estudo?

A presente proposta sugere um sistema intuitivo onde facilmente seja possível encontrar a origem de problemas. Conseguem-se encontrar ferramentas que mostram de forma clara qual o estado dos componentes físicos de cada nó. Existem outras ainda que oferecem o estado dos componentes lógicos, como representação de *logs*, etc. No entanto, a sua representação não deve ter como objetivo única e simplesmente mostrar ao utilizador qual o estado destes componentes. O sistema deve possuir várias opções de representação dos dados, onde seja possível retirar conclusões acerca de relações entre as variações de desempenho. Será fácil ao utilizador retirar dos atuais sistemas de monitorização conclusões acerca das possíveis consequências de uma troca de funções entre máquinas? De forma a auxiliar essas conclusões, o sistema deve oferecer várias representações da rede, com relações e partilha de recursos, por forma a facilitar a perceção da organização dos componentes.

Um sistema de monitorização tem a capacidade de recolher, processar e de representar os dados monitorizados de forma visual. De acordo com Edmund Wong [Won], "*network monitoring techniques are developed to allow network management applications to check the states of their network devices*." No entanto, a sua representação pode não ser simples. Os utilizadores podem ter preferências e pontos de vista diferentes. Como Chris Blake [CM] afirma "*Why is there no EASY and SIMPLE monitoring program out there that is Open Source, customizable, AND functional?*"

". O principal objetivo desta dissertação é apresentar uma proposta de solução que componha um sistema de monitorização capaz de resolver problemas de escalabilidade no que diz respeito à recolha de dados de monitorização e apresente uma interface gráfica coerente com esta escalabilidade, capacitando os utilizadores com uma ferramenta capaz de representar os componentes do sistema e as suas respetivas relações.

1.4 Estrutura da Dissertação

O presente capítulo apresenta a contextualização e motivação do estudo realizado, bem como os problemas que se pretendem resolver. O capítulo 2 fornece informação acerca do estado da arte do objeto de estudo, uma revisão das técnicas de visualização usadas, uma revisão das ferramentas de monitorização com, atualmente, maior impacto e as aplicações que as complementam. O capítulo 3 apresenta o problema principal e a proposta de uma solução, mencionando quais os requisitos, imposições e casos de uso exigidos. O capítulo 4 aborda os detalhes da implementação da solução, nomeadamente a implementação da representação da estrutura do sistema, desenvolvimento e escolha das melhores alternativas gráficas e modelação dos *dashboards*. Posteriormente, o capítulo 5 detalha a implementação da solução, composta por várias etapas. Entre elas, destacam-se a avaliação do modo de comunicação Gânglia, a arquitetura proposta, modelos de dados, escolha de tecnologias de desenvolvimento, estudo de uma interface sob forma de maquete, diagrama de fluxo exigido face aos casos de uso existentes e detalhes de desenvolvimento modular. O capítulo 6 apresenta os resultados obtidos após testes da solução implementada. Por fim são apresentadas as conclusões (capítulo 7) do trabalho realizado e possível trabalho futuro no sentido de robustecer ainda mais a solução proposta. Bibliografia e anexos são também capítulos integrantes no final do presente documento.

Capítulo 2

Revisão Bibliográfica

O presente capítulo apresenta a revisão do estado da arte dos sistemas de monitorização e das ferramentas que os complementam. São apresentadas as técnicas de visualização mais relevantes para o objeto de estudo, seguido de alguns exemplos de ferramentas de monitorização. Por fim, são apresentadas ferramentas tecnológicas de relevo para este domínio e mencionados os seus principais benefícios.

2.1 Técnicas de visualização

Acompanhadas das técnicas de representação dos sistemas de monitorização, estão associadas técnicas de representação visual de dados. Este capítulo refere algumas das técnicas de visualização que podem ser úteis na nova proposta de representação de dados do sistema de monitorização.

Antes de se perceber quais as características que um bom sistema de monitorização deve ter é importante saber qual a razão do seu uso como meio de examinação de dados. Como defendem Noah Iliinsky e Julie Steele [IS11], a representação de dados através de gráficos apresenta uma elevada capacidade de transportar informação para o cérebro e é um excelente método de exploração visual que pode despontar novas questões e assim tornar-se em uma excelente forma de identificar problemas.

De acordo com [Wik15c] um diagrama é definido como uma ilustração esquemática de um conjunto de dados. A par dessa ilustração existe um conjunto vasto de técnicas de representação visual que podem aumentar a eficácia de comunicação dos dados. Nico Marrero argumenta em [Mar07] que, apesar do estudo dessas técnicas estar ainda na sua infância, a representação gráfica de dados já é usada há muito tempo. Um exemplo é o uso do Abacus pelos nossos antepassados: uma ferramenta de cálculo visual que auxiliava as transações comerciais, definido na página [Wik15a].

Existem dois grandes padrões de representação visual: representação de dados e representação de informação. De acordo com Edward Tufte e Graves-Morris [TGM83] existem mais desafios na

representação de dados do que na representação de informação, devendo-se ao facto da representação dos dados estar associada ao desenho através de algoritmos, pois têm de saber lidar com diferentes dimensões de dados, desconhecidas previamente. Um sistema de monitorização refere-se à representação de dados, devendo estes portanto, ser desenhados através de algoritmos de análise. Existem dois objetivos diferentes na representação de dados: explorar e explicar. Normalmente, a explicação de dados está associada ao objetivo de contar uma história, onde conclusão deve ser objetiva. Por outro lado, a exploração de dados é usada quando se pretende descobrir muitas histórias, ou seja, várias conclusões, e quando o tipo de dados é previamente desconhecido.

Os dados de monitorização são uma conjugação destas duas categorias. Por um lado, o autor sabe qual o tipo de dados a ser representado, no entanto, o sistema tem de ser capaz de se adaptar para conseguir representar os dados em diferentes estruturas.

Na representação de dados estão normalmente envolvidas três entidades: o leitor, os dados e o autor (designer, pessoa que cria os mecanismos de representação de dados). A interação entre cada uma destas entidades contribui para uma propriedade importante na visualização de dados, segundo [TGM83].

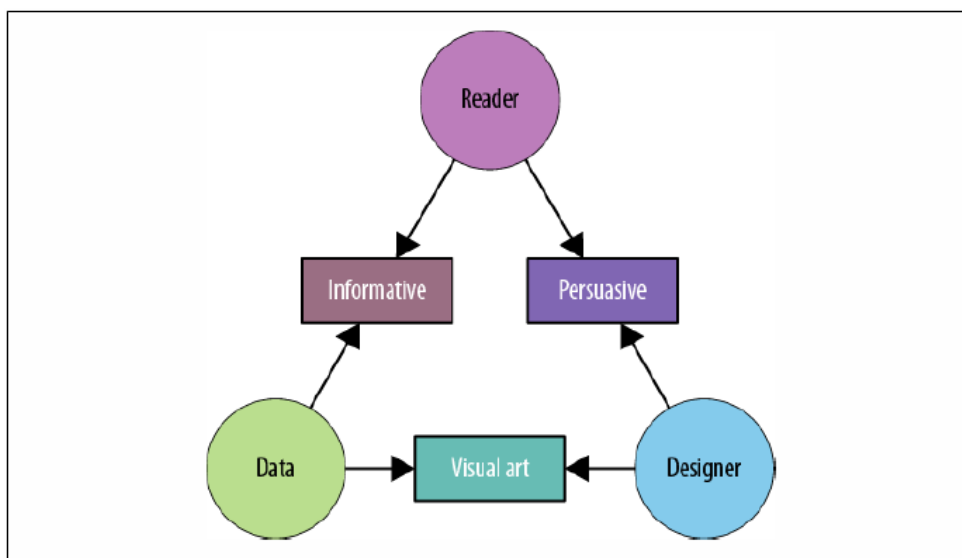


Figura 2.1: Triângulo de visualização

Como é possível observar na figura 2.1, o leitor interage com os dados com uma finalidade bem definida: obter a informação pretendida. A arte visual é a propriedade partilhada pelo designer e pelos dados. Esta partilha deve-se ao facto da comunicação visual dos dados ter de ser eficaz, transformando dados brutos em informação visual. A transmissão de informação não contém objetivos limitados. Existe também a necessidade de mudar a forma como o leitor "vê" os dados, inculcando nele rotinas de análise e criando uma relação persuasiva entre leitor e *designer*.

Quando alguém pretende transmitir informação é necessário que se coloque na pele do recetor. Novamente, Edward Tufte e Graves-Morris [TGM83] afirmam que *"If we agree that the purpose of the visualization is to take a story that is already known to you and tell it to somebody else,*

then it stands to reason that the somebody else is exactly that—not you, but another ". Portanto, o primeiro desafio é identificar o público a quem queremos contar a história e depois usar a linguagem apropriada à comunicação. Características como a cor e o uso de ícones apropriados devem ser estudados. Depois, é importante conhecer a estrutura principal dos dados: se são temporais, com estrutura hierárquica, quantas variáveis têm, qual a sua relação, etc. Andries Van Dam et al. defendem em [VDFL⁺00] que a visualização é um método poderoso usado em ferramentas de monitorização porque explora o melhor canal perceptual humano: a visão.

Mais de 50% dos neurónios do cérebro humano são dedicados à visão, o que resulta em excelentes capacidades de reconhecimento de padrões. De acordo com Daniela Petrelli et al. em [PMDC09], a visualização de informação não tem apenas a responsabilidade de disponibilizar dados, mas também desenvolver capacidades perceptuais e cognitivas. O mapeamento de atributos de dados para características visuais é uma forma de se cumprir o objetivo. Christopher Healey [Hea96] defende: *"Color is an important and frequently-used feature. Examples include color temperature gradients on maps and charts, color-coded vector fields in flow visualization, or color icons displayed by real-time simulation systems."* Algumas técnicas de manipulação de layout, tais como zoom ou transformação de componentes são também úteis para focar a atenção do leitor e construir pontos de interesse, como defendem Andreas Buja et al. [BMMS91] e Mackinlay et al. em [MCR90]. Shneidermann [Shn96] afirma: *"overview first, zoom and filter, then details-on demand"*. A ideia principal prende-se ao facto de fornecer ao leitor uma vista geral do sistema, com foco progressivo nos aspetos a realçar. Este reconhecimento comprova a estrutura centrada no utilizador proposta por Dadzie et al. [DLP09] e visível na figura 2.2.

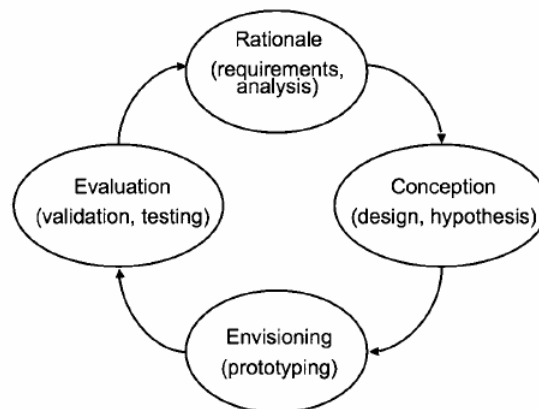


Figura 2.2: Avaliação da estrutura centrada no utilizador

Todos estes aspetos são importantes para facilitar a decodificação da informação pelo utilizador, aumentando a disponibilidade cerebral para análise dos dados. De acordo com Edward Tufte et al. [TGM83], *"When reading a visualization (or any other kind of communication), the reader has a limited amount of brainpower to dedicate to the problem (figura 2.3). Some of this brainpower will be dedicated to decoding the visualization; any brainpower that is left may then be used to*

understand the message (if the reader has not yet given up in frustration). In order to be successful, your job is to facilitate understanding by minimizing the amount of extraneous searching and decoding necessary to get the message. "

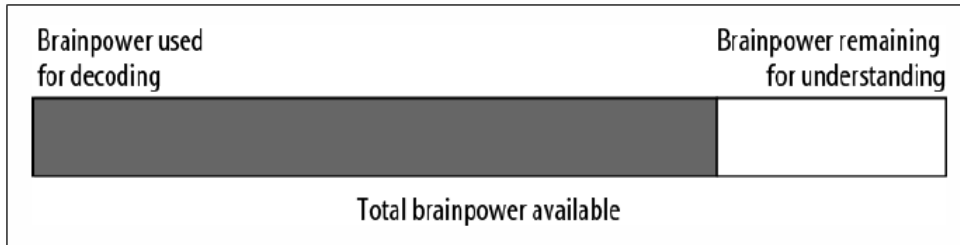


Figura 2.3: Disponibilidade cerebral na análise dos dados

2.2 Sistemas de Monitorização

Tal como referem Sreeja e Chaudhari [SC15], os sistemas de *Big Data* operam geralmente em sistemas de rede que acedem a recursos distribuídos, onde circulam dados heterogéneos. De acordo com Sergio Andreozzi et al. [ADBF⁺05], os sistemas distribuídos possibilitam a coordenação de recursos e serviços, ao contrário de mecanismos centralizados. Consequente dessa coordenação de recursos existem variações comportamentais ao longo da execução das tarefas, alterando o desempenho individual de cada máquina. Tian [Xu11] refere, que existem ferramentas capazes de recolher essas métricas de análise e disponibilizá-las em formato gráfico, as ferramentas de monitorização. O tratamento dessas métricas é um processo essencial. De acordo com Jing Xia et al. [XWG⁺13], estes *clusters* originam um enorme volume de *logs* com baixo valor informativo. É necessária uma análise destes para que o output tenha maior valor e seja capaz de fornecer dados com maior potencial e ajude a tomar decisões críticas em tempo útil. O comportamento destes sistemas potencialmente escaláveis é também um fator crítico para a existência de uma ferramenta que monitorize a sua atividade, defendido por Tian [Xu11]. Já Sreeja e Chaudhari [SC15] mencionam como principal vantagem de um sistema de monitorização o facto de possuir um sistema de deteção de falhas e sugestão de mecanismos de recuperação. A construção de um sistema de monitorização está normalmente dividida em duas etapas:

- Fase de processamento, onde o sistema recolhe de cada nó/componente monitorado o valor das métricas em análise e as armazena numa estrutura de base de dados de acesso estável;
- Fase de visualização, onde o sistema recolhe os valores da base de dados, transforma-os e os disponibiliza em formato visual.

Weber e Thomas definem as ferramentas de monitorização em [WT06] como ferramentas de indicação de performance (*Key Performance Indicator*). Silva e Catarci [SC00] defendem que todos os dados recolhidos por sistemas de monitorização são caracterizados por serem dados lineares e temporais, vulgarmente representados em gráficos de linhas. Este tipo de gráficos consegue

relacionar de forma eficaz o seu comportamento com o tempo decorrido. O primeiro passo é identificar os tipos de dados e classificá-los. De acordo com Herman et al. [HMM00], os dados sem estrutura definida são aqueles que não variam com a alteração de outros, tornando-se impossível relacioná-los.

2.2.1 Sistemas de monitorização

A presente secção apresenta uma revisão dos principais sistemas de monitorização, analisando as suas principais características e limitações.

2.2.1.1 Ganglia

O Ganglia é das mais completas ferramentas de monitorização, responsável pela recolha, armazenamento e disponibilização de dados de monitorização. De acordo com a página oficial [Gan01], "*Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and grids*". A principal vantagem deste sistema prende-se ao seu método de recolha de dados, recorrendo a rotinas em tempo real em cada ponto de análise, que comunicam com a recolha através do sistema operativo de forma distribuída ou centralizada. O Ganglia foi desenvolvido com o objetivo de sustentar milhares de máquinas partilhando informação a cada 15 segundos, sendo esta a principal vantagem face às demais ferramentas. Segundo Andreozzi et al. [ADBF+05], o Ganglia apresenta uma estrutura hierárquica composta por três elementos: *gmond*, *gmetad* e *ganglia web client*.

O *gmond* é responsável pela recolha dos valores em análise em cada máquina, interagindo diretamente com o sistema operativo, seguindo regras de escalonamento definidas em ficheiros de configuração. Cada elemento da rede partilha os seus dados em formato XDR, usando protocolos *multicast* ou *unicast*. O protocolo *multicast* oferece, como vantagens, o facto de todos os elementos integrantes do sistema ficarem com conhecimento dos dados trocados. No entanto, e segundo Neves et al. [NSC05], este método pode originar graves problemas de excesso de fluxo na rede.

O *gmetad* é o componente que estabelece a ponte entre a recolha de dados transmitidos pelo *gmond*, através da rede, e o armazenamento em formato *round robin database* (RRD). Este formato tem como conceito a alocação estática de valores sob vários *chunks*. Massie et al. [MCC04] defendem que este formato é muito usual em sistemas de armazenamento de dados temporais, especialmente pelo facto do seu tamanho se manter constante ao longo do seu tempo de vida.

O componente responsável pela representação gráfica dos dados é o *ganglia web client*. Este componente efetua chamadas a uma API que tem a função de gerar gráficos em formato imagem a partir de uma base de dados RRD. Os gráficos gerados por este componente podem ser vistos da figura 2.4. As principais funcionalidades do frontend do Ganglia são:

- Listagem das máquinas organizadas por hierarquia;
- Visualização de gráficos de linhas de métricas selecionadas. Por exemplo, é possível obter um resumo de todas as métricas de uma máquina;

Revisão Bibliográfica

- Campo de pesquisa;
- Capacidade de criar vistas com gráficos de diferentes origens;
- Agregar várias métricas a um gráfico, funcionalidade útil na comparação de várias métricas;
- Comparação de dados entre máquinas, criando automaticamente gráficos com métricas em comum;
- Eventos: Linhas verticais disponibilizadas em gráficos de linhas que definem um instante de tempo;
- Reportar um valor de análise;
- *Dashboards* com troca automática de gráficos;
- *Dashboards* personalizados.

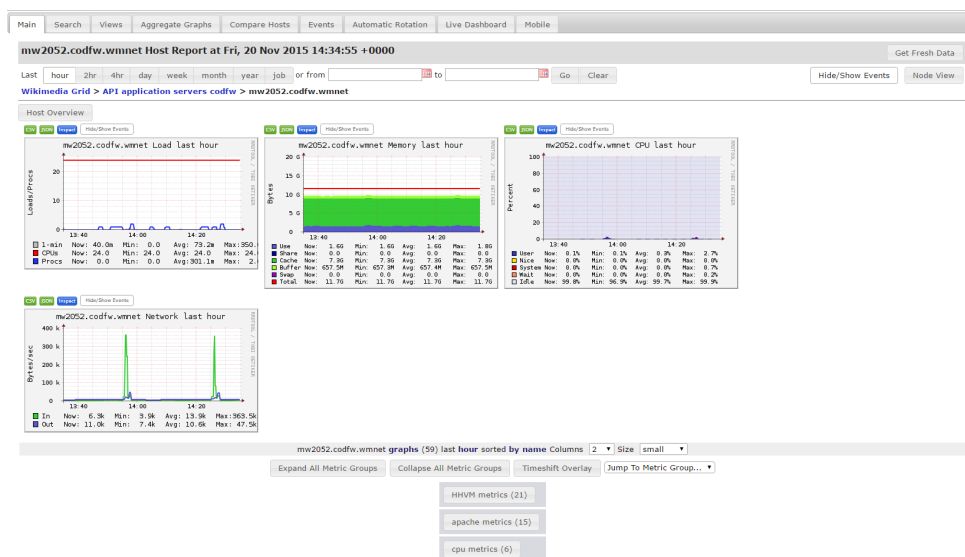


Figura 2.4: Pormenor da interface Ganglia

2.2.1.2 Observium

De acordo com a página oficial [Doc], "*Observium is an auto-discovering network monitoring platform supporting a wide range of hardware platforms and operating systems (...) seeks to provide a powerful yet simple and intuitive interface to the health and status of your network*". Este sistema de monitorização usa SNMP para recolha e partilha de informação dos dispositivos, onde é posteriormente centralizada para apresentação em plataforma web. Os dispositivos podem ser máquinas, *routers* ou até terminais, sendo sempre comandados por um sistema central de controlo, que faz com que, segundo Blake [Bla13], o Observium se torne um bom sistema candidato para

Revisão Bibliográfica

quem não pretende gastar muito tempo com configurações, tendo uma capacidade de auto descobrir o sistema, construindo a sua estrutura e gerando a interface de forma automática, fator que o diferencia da concorrência. No entanto, este automatismo tem limitações quando se depara com DNS e apresenta limitações de configuração.

Outra potencialidade do Observium prende-se com a sua interface, representada na figura 2.5, que apresenta uma representação visual muito intuitiva, habilitando uma pesquisa em profundidade objetiva, onde se pode organizar os componentes por grupos e fazer facilmente fazer comparações entre máquinas. O Observium está também habilitado de um sistema de alertas, ótimo para definir valores limites em cada parâmetro.

À semelhança do Ganglia, o Observium armazena os dados recolhidos sob formato RRD, usando o pacote RRDtool para interação com a base de dados. No entanto não está preparado para trabalhar num sistema escalável e não permite a adição de mais componentes de análise.

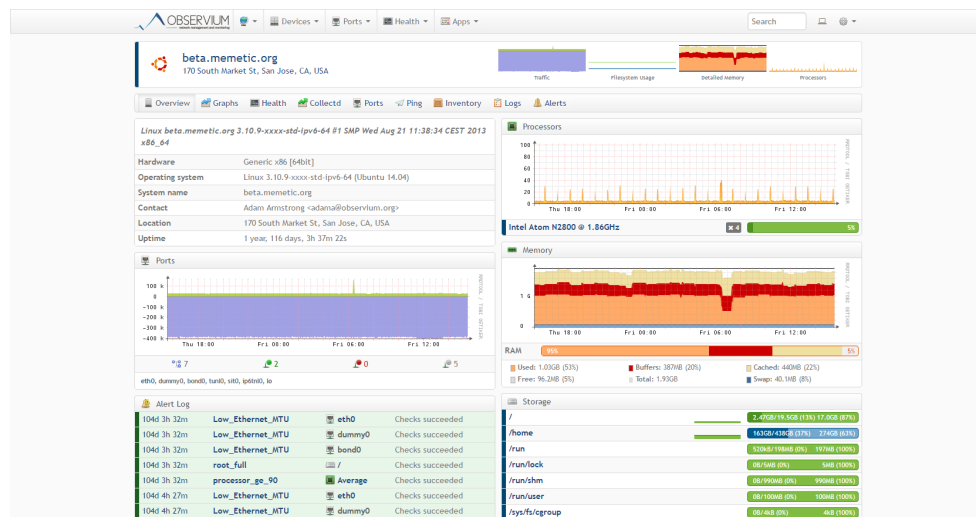


Figura 2.5: Pormenor da interface Observium

2.2.1.3 Nagios

O Nagios é um sistema de monitorização de código aberto. A sua página oficial [Sol09] refere que possui as normais funcionalidades encontradas nesta classe de sistemas, designadamente, serviços de rede, captura de dados sobre recursos em cada nó, envio dos dados para um sistema centralizado e disponibilização através de uma interface gráfica. Apesar de se identificar como uma ferramenta amigável do utilizador e fácil de trabalhar, possui muitas funcionalidades desnecessárias. Segundo Andreozzi et al. [ADBF⁺05], o Nagios apresenta a grande desvantagem de apenas funcionar em redes locais (*Local Area Network*). No entanto, Joseph [Jos15] refere que o Nagios possui um excelente sistema de alertas locais, via correio eletrónico ou via móvel.

2.2.1.4 MonALISA

MonALISA [Mona] é um sistema de monitorização produzido pela Caltech que fornece gráficos lineares ou circulares representativos dos dados em série. No entanto, o que faz sobressair este sistema é a representação espacial do sistema, através da localização georreferenciada de cada componente. Segundo Andreozzi et al. [ADBF⁺05] este sistema é baseado em agentes e fornece uma completa monitorização de aplicações, serviços, tráfego e medidas de desempenho. Algumas das métricas, referidas por Legrand et al. [LNV⁺09], são o consumo de CPU, o tempo de execução, o tamanho disponível de *cache*, a frequência dos cores, o uso de memória, ficheiros abertos, percentagem de utilização do CPU, entre outros. O seu principal objetivo, segundo Newman et al. [NLG⁺03] é o de fornecer dados de monitorização a serviços de mais alto nível, para melhoria de performance.

2.2.1.5 MapCenter

O MapCenter é um sistema de monitorização um pouco mais limitado do que as soluções anteriormente, pois apenas analisa desempenho de serviços e aplicações na rede e não sobre outros recursos (CPU, memória, etc.). De acordo com Bonnassieux et al. [BHP02], o seu maior objetivo é fornecer informação acerca da disponibilidade de serviços na rede. À semelhança da ferramenta MonALISA, também oferece uma representação geoespacial do sistema monitorado, com representação das ligações entre os serviços e aplicações. A sua arquitetura é composta por três camadas: Monitorização: Recolhe dados dos sensores de recursos; Armazenamento: Armazena internamente os dados; Apresentação: Fornece vistas diferentes da estrutura.

A camada de apresentação tem qualidades que merecem destaque, como a capacidade de atualização com intervalos personalizados e a representação global da estrutura. Bonnassieux et al. [BHP02] refere três vistas principais: a vista em árvore, com representação do estado de cada nó; a vista lógica, com uma representação semelhante à anterior mas mais abstrata; a vista geográfica, já mencionada anteriormente.

2.2.1.6 Cacti

O Cacti é um sistema de monitorização de código aberto, escalável (semelhante ao Nagios ou Ganglia) que utiliza o sistema de armazenamento RRD, como afirmado Swinney [Swi12]. A sua interface de apresentação de dados monitorizados baseia-se em ambientes web, onde o acesso é restrito e gerido sob direitos de acesso mediante o tipo de utilizador. No entanto o Cacti não possui nenhum tipo de sistema de alertas próprio, existindo a necessidade de o integrar com outras ferramentas. A equipa do Cacti [Gro] afirma que "*Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box.*".

2.2.1.7 Zabbix

O Zabbix é mais um sistema de monitorização, escalável e distribuído. No entanto, o Zabbix não usa o RRD como formato de armazenamento (apesar de não ser considerada uma desvantagem). Este sistema tem mecanismos capazes de recolher dados de protocolos como HTTP, FTP, SSH, POP3, SMTP, SNMP, MySQL, etc. Como o Observium, tem a capacidade de auto descobrir o seu meio e representar a estrutura mediante este. Adicionalmente, o Zabbix apresenta funcionalidades como *JMX monitoring*, autenticação e permissões baseadas em regras, personalização de notificações e registo de logs de atividade. No entanto, segundo Joseph [Jos15], "*As it is more complex in design and interface you will take some time to get in touch with this tool.*". Opinião partilhada por Simmonds e Harrington em [SH09], que defendem que apesar da sua interface web com aspeto visual atrativo, tem uma configuração pobre baseada em importação de ficheiros XML.

2.2.1.8 VisPerf

O VisPerf é definido por Gerndt et al. [GWB⁺04] como um bom sistema de monitorização e visualização. Já Lee et al. [LDR03] defendem que este sistema tem bons mecanismos de recolha de dados, no entanto com uma visualização pobre, comprovada na figura 2.6. O mecanismo de recolha de dados pode ser executado através da análise de ficheiros de log ou de serviços diretos e são disponibilizados através de uma API intermédia. Sreeja [SC15] refere que, além de monitorização local, o VisPerf também efetua monitorização remota. No entanto não tem nenhuma representação visual da estrutura.

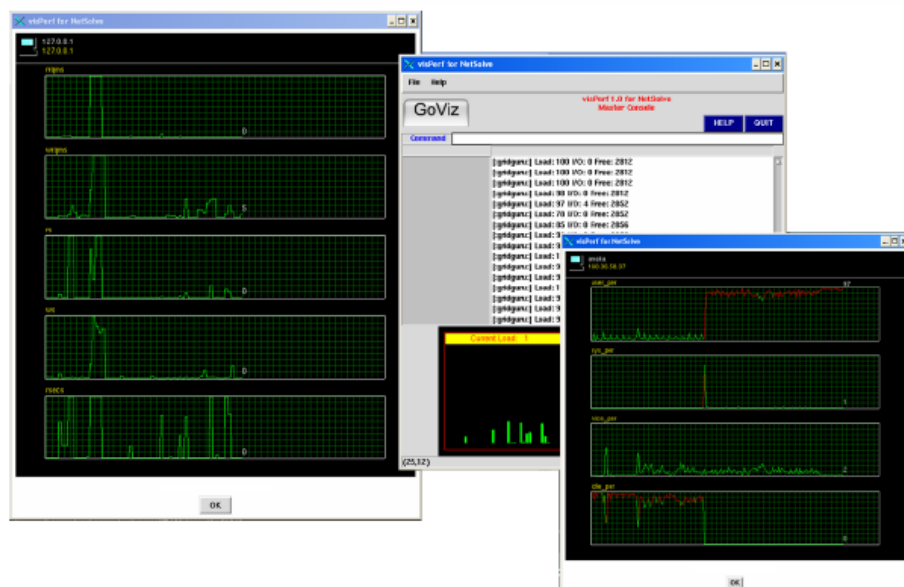


Figura 2.6: Pormenor da interface VisPerf

2.2.1.9 Conclusão

Na tabela 2.1 é apresentado um resumo das principais características dos sistemas analisados, nomeadamente se são de código aberto, a sua potencial escalabilidade, o método de recolha de dados e se contêm representação georreferenciada e sistema de alertas.

Tabela 2.1: Resumo das características das ferramentas analisadas

| Nome | <i>Open source</i> | Escalabilidade (quantidade de dispositivos) | Representação georreferenciada | Sistema de alertas | Método de recolha de dados |
|-----------|--------------------|---|--------------------------------|--------------------|----------------------------|
| Ganglia | Sim | Milhares | Não | Não | gmond |
| Observium | Não | Dezenas | Sim | Sim | SNMP |
| Nagios | Sim | Dezenas, apenas em redes locais | Sim | Sim | NSClient++ |
| MonALISA | Não | Centenas | Sim | Não | SNMP |
| MapCenter | Não | Não especificado | Sim | Não | MDS |
| Cacti | Sim | Dezenas, apenas em redes locais | Não | Não | SNMP |
| Zabbix | Não | Milhares | Sim | Sim | SNMP |
| VisPerf | Não | Dezenas | Não | Não | logs e eventos |

2.2.2 Ferramentas de visualização

Nesta secção apresentam-se duas ferramentas - Graphite e Grafana - que complementam alguns dos sistemas apresentados na secção (2.2.1). Não são sistemas de monitorização porque não recolhem dados de monitorização. No entanto, desempenham um papel relevante na representação visual de dados.

2.2.2.1 Graphite

O objetivo principal do Graphite é o de converter os dados presentes nos ficheiros RRD (ou em ficheiros *gzipped whisper*) em representações gráficas, como se pode observar no exemplo da figura 2.7. O Graphite disponibiliza serviços de rede através do protocolo HTTP, sendo essa a principal vantagem face a outras ferramentas de conversão de RRD, capacitando-o para ser utilizado em sistemas de alta escalabilidade com acesso em tempo real. De acordo com Chris Davis [Dav], "One of the most common uses of Graphite is building web-based dashboards for monitoring and analysis."

Revisão Bibliográfica

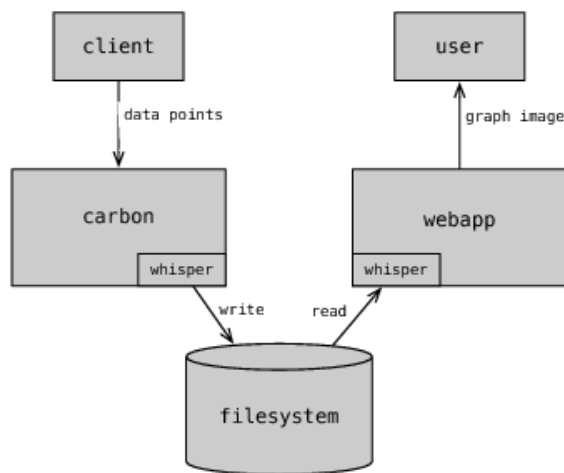


Figura 2.7: Fluxo de dados no Graphite

Ellingwood apresenta o Graphite, em [Ell14], como uma ferramenta composta por três principais componentes: Whisper, uma biblioteca especializada em controlo de dados, mantendo os dados de forma paralela com RRD, mas com maior velocidade de acesso; Carbon, componente que está à escuta de outros agentes e tem funções de *caching* para otimização I/O; Aplicação cliente, ilustrada na figura 2.8 que, simultaneamente, pode fornecer os seus dados através de uma API.

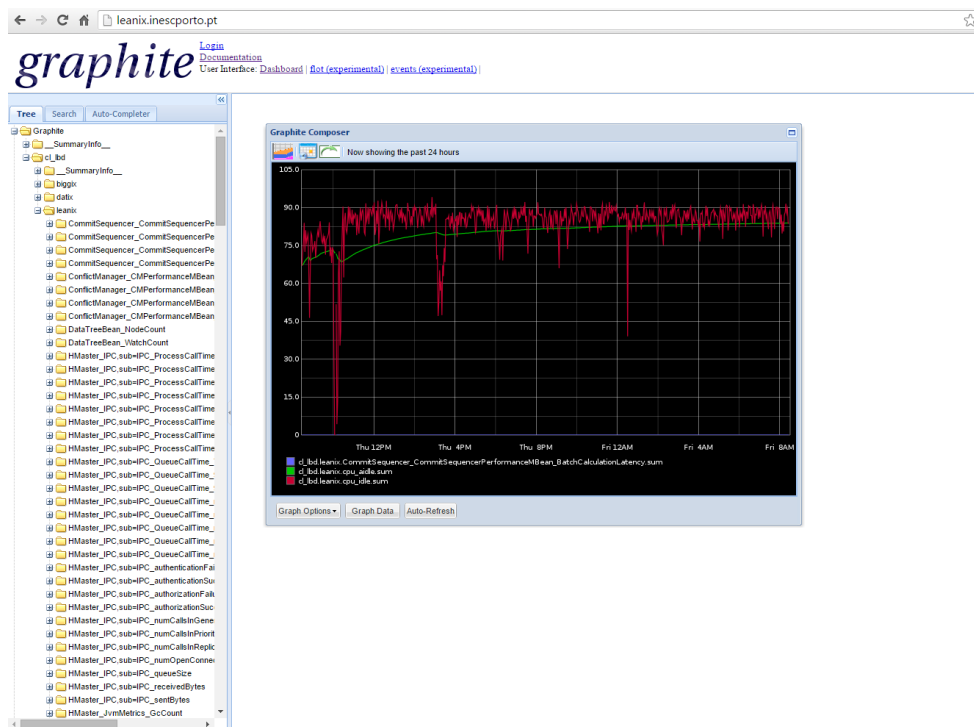


Figura 2.8: Pormenor da interface Graphite

2.2.2.2 Grafana

Grafana é uma ferramenta de representação gráfica de dados de monitorização que apresenta *dashboards*, conforme ilustra a figura 2.9. O Grafana permite a total manipulação e ordenação de elementos numa interface muito user-friendly e responsiva. Ödegaard [Öd15] define o Grafana como uma aplicação com o objetivo de facultar dados em séries temporais para aplicações de análise. Com o Grafana é possível:

- Personalizar totalmente cada elemento de um *dashboard*, incluindo o nome, cores e estilos, adicionar ou remover funções e mudar tempos de atualização;
- Personalizar totalmente o *dashboard* em si, incluindo tamanhos, posições, ordenação e tempos de atualização globais a todos os gráficos.

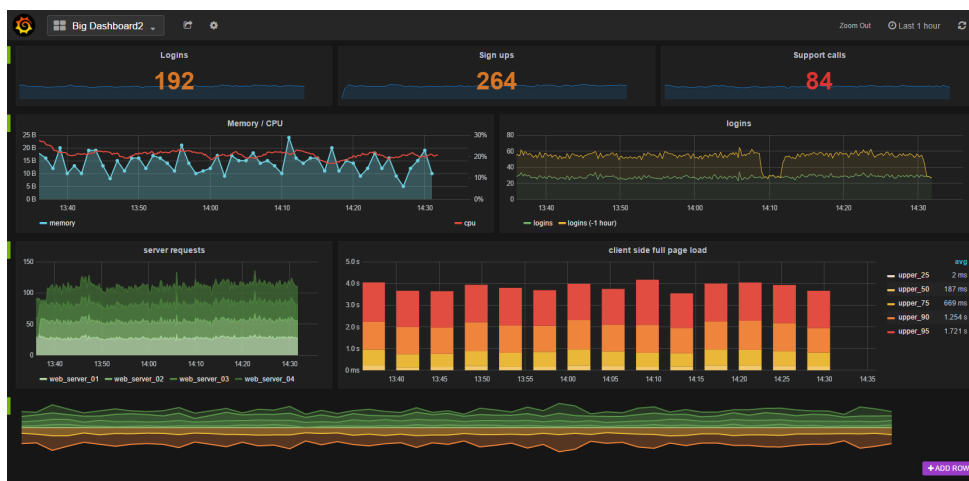


Figura 2.9: Pormenor da interface Grafana

2.3 Tecnologias de visualização

Nesta secção apresentam-se tecnologias estudadas que se consideram adequadas ao desenvolvimento ou extensão de um sistema de monitorização, ao nível da representação visual de dados.

2.3.1 D3.js

O D3.js é uma biblioteca *open-source* Javascript para manipulação de dados que usa HTML, SVG e CSS para gerar gráficos. Esta biblioteca usufrui de excelentes capacidades contidas nos *browsers* atuais sem a necessidade de construir uma ferramenta de origem, combinando excelentes componentes visuais com uma manipulação de DOM extraordinária. Sendo assim, o D3.js é capaz de transformar dados arbitrários em Document Object Model (DOM) e posteriormente aplicar-lhe modificações. Por exemplo, pode ser gerada uma tabela HTML a partir de uma lista de números

e a qualquer momento usar essa mesma lista para criar um gráfico de barras. O D3.js permite gerar qualquer tipo de animação visual, possibilitando uma infinita conjugação de potencialidades, originando qualquer tipo de representação gráfica. Segundo Mike Bostock em [Bos11], esta biblioteca tem o mínimo de *overhead* esperado, sendo extremamente rápida e com um grande suporte a estruturas de dados.

2.3.1.1 NVD3

O NVD3 é uma ferramenta de suporte à criação gráfica em Javascript, tal como segue a página [Par]. Esta ferramenta disponibiliza gráficos com o auxílio da biblioteca D3.js. O NVD3 oferece gráficos de linhas, barras, dispersão, circulares e bullet, com variadas personalizações de cores, nomes, descrições e legendas. A ferramenta apresenta-se como uma ótima extensão ao D3.js, possibilitando a criação de gráficos através de mecanismos já desenvolvidos.

2.3.2 Chartlist.js

Chartlist.js é outra biblioteca open-source Javascript de construção de gráficos, disponível em [Kun], por Gion Kunz. Apresenta vantagens como o facto de possuir vários tipos de gráficos pré-definidos, como linhas, barras, circulares e *gauge*, todos responsivos, qualidade imprescindível para a construção de *dashboards*. No entanto, a variedade de gráficos não é expansível, pois a ferramenta apenas permite o uso dos gráficos existentes.

2.3.3 Google Charts

O Google Charts é mais uma biblioteca de construção de gráficos com mais variedade que a biblioteca anterior. Gráficos geográficos, gráficos de pontos, histogramas, gráficos de barras, gráficos de linhas, de bolhas e ainda *treemaps* são só alguns exemplos das possibilidades que esta oferece. No entanto, à semelhança do Chartlist.js, esta biblioteca permite instanciar tipos de gráficos já existentes, sendo impossível a construção de novos tipos.

2.3.4 Gridster

O Gridster, segundo a página [Duc] é uma biblioteca de desenvolvimento em ambientes Javascript para gestão de *dashboards*. Esta biblioteca oferece manipulações dinâmicas dos elementos presentes, tanto em dimensão como em posição. Apesar da liberdade de manipulação oferecida, os *dashboards* são limitados a dimensões de seis unidades de largura, sendo que cada elemento tem de assumir dimensões e posições inteiras.

2.4 Sumário

O presente capítulo focou-se nos sistemas de monitorização, as suas qualidades e desvantagens. Além dos sistemas analisados existem outros que possuem boas características, como o

QlikView [Sol], Pandora FMS [Sys], entre outros. No entanto, é difícil encontrar um sistema com todas as características ideais de captura, partilha, armazenamento e também com uma interface visual fácil e intuitiva que ajude o utilizador a tomar decisões. Usando dados da página [Wik15b], pode-se observar que a ferramenta Observium possui um *front-end* com excelentes características, no entanto, não tem a capacidade de lidar com sistemas em rede e muito escaláveis. Pelo contrário, o Ganglia oferece um excelente sistema que cumpre em pleno com a primeira parte das tarefas, no entanto o seu cliente web está longe de ser o mais aconselhável.

Em seguida foram analisadas duas ferramentas de visualização que têm o papel de acompanhar aplicações de monitorização. Conclui-se que o Graphite é uma excelente opção de conversão de dados, caso seja utilizados ficheiros RRD como formato de armazenamento. Já o Grafana é uma ótima ferramenta de gestão de dashboards, onde são verificadas as funcionalidades mais requeridas em processos como este.

Por fim foram analisadas bibliotecas de representação gráfica suportadas na linguagem Javascript, com destaque para o D3.js. Esta biblioteca oferece mecanismos de construção gráfica, funcionalidade perfeita para a criação de novos gráficos de estrutura, difíceis de encontrar em bibliotecas deste género.

O próximo capítulo especifica os problemas presentes nas ferramentas estudadas e de que forma podem ser melhorados para a construção de uma nova solução. São também realçadas algumas características que já fazem parte desses sistemas e que devem fazer parte da solução proposta no presente estudo.

Capítulo 3

Descrição do problema

Este capítulo descreve em detalhe o problema encontrado nos sistemas de monitorização estudados, realçando os requisitos e especificações consideradas relevantes para a proposta de um sistema de monitorização inovador.

3.1 Limitações dos sistemas de monitorização

Com base na análise dos sistemas que existem atualmente no mercado é possível concluir que existem aspetos que podem ser melhorados, de forma a melhor resolver necessidades dos utilizadores. Os principais problemas encontrados são:

- Oferta de tipos de representação gráfica reduzida, uma vez que quase todos os sistemas limitam-se a representar os dados através gráficos de linhas;
- Pouca personalização gráfica, no que diz respeito a cores, títulos, descrições e legendas;
- Atualização de gráficos corresponde a transferir imagens, exigindo novas imagens para cada atualização;
- Representação limitada da estrutura do sistema monitorizado, baseando-se em listas de navegação em profundidade, o que dificulta a localização de componentes;
- Representação orientadas aos componentes físicos do sistema monitorado e não à interpretação lógica dos serviços existentes nesse sistema;
- *Dashboards* com manipulação reduzida;
- Organização complexa e por vezes de difícil interpretação.

3.2 Proposta de solução

Tendo em conta os problemas identificados, o presente trabalho define como principal prioridade o estudo de melhoramentos ao nível do aspeto visual da aplicação. O primeiro objetivo passa por representar de forma gráfica a estrutura da rede, transmitindo ao utilizador uma boa perceção dos componentes constituintes do sistema e a relação entre estes.

Assim, é proposta uma solução onde o principal objetivo é fornecer ao utilizador a possibilidade de identificar, não só que existe um problema, mas a origem deste. Para esse efeito é essencial que sejam oferecidas várias opções de representação do sistema monitorado e dos serviços disponibilizados, possibilitando a rápida geração de *dashboards* para análise em tempo real. A disponibilização de *dashboards* apropriados para cada componente do sistema, pode também ajudar a tornar a exploração mais rápida e eficaz.

Apesar de se pretender uma solução intuitiva, considera-se que o utilizador deve ter a literacia adequada à monitorização de sistemas distribuídos, sendo capaz de analisar as consequências da variação de uma métrica. O utilizador deve ainda possuir conhecimentos de análise de desempenho de redes, para que seja capaz de retirar proveito das vantagens que a solução proposta lhe possa oferecer.

No entanto, um sistema de monitorização não é só composto pelo seu aspeto visual; é necessário encontrar um sistema que cumpra com os requisitos do sub-sistema de *backend* e tenha a capacidade de disponibilizar os dados através de protocolos de comunicação previstos nesse sub-sistema. Conforme analisado no capítulo anterior e em consonância com os requisitos apresentados, o Ganglia apresenta todas as características necessárias e importantes para a realização das tarefas exigidas ao nível da recolha e do armazenamento de dados de monitorização. Este sistema disponibiliza métodos de análise lógica, tornando possível a adição de métricas de análise orientadas aos serviços presentes na rede.

A opção pela componente de servidor do Ganglia faz com que a solução proposta possa manter os elevados níveis de escalabilidade na recolha e armazenamento de dados. Após a realização de alguns testes comportamentais ao Ganglia, conclui-se que este sistema apresenta as qualidades ideais para a realização das tarefas de *backend*. Por conseguinte, a solução de cliente web proposta neste estudo utiliza um sistema Ganglia na recolha e no armazenamento de dados e necessita de comunicar com os serviços existentes no Ganglia. A avaliação da solução proposta está sujeita a testes em ambientes de produção, sendo que para efeitos de teste, é utilizada uma instância do Ganglia de pequenas dimensões (três máquinas). A solução proposta terá de usufruir de excelentes mecanismos de modelação, sendo capaz de integrar uma solução aplicacional já existente, recorrendo a partilha de modelos de dados e bibliotecas de auxílio ao desenvolvimento.

Como referido anteriormente, apesar das suas vantagens de *backend* referidas, constatou-se que o Ganglia apresenta debilidades ao nível do *frontend* (resultantes das limitações da tecnologia disponível na altura em que foi desenvolvido), nomeadamente o facto de criar os gráficos em formato de imagem, do lado do servidor, e posteriormente enviá-los para o cliente. Com isto, facilmente se percebe que:

Descrição do problema

- Imagens PNG consomem uma largura de banda demasiado elevada quando comparada com a transmissão de dados em formato de texto;
- Com a variação de valores resultantes das atualizações sucessivas a cada 15 segundos provenientes do Gangleia, a geração de um novo gráfico e o seu envio pela rede, faz com que haja a necessidade de repetição de informação enviada, uma vez que apenas seria necessário o envio do valor de atualização.

Na solução proposta, considera-se que a construção dos gráficos deve ser feita do lado da aplicação cliente, para que haja uma redução substancial na quantidade de informação a circular entre o servidor e o cliente web.

Adicionalmente, do levantamento efetuado, conclui-se que existem boas características ao nível da interface do cliente web, a reter na solução proposta, nomeadamente:

- Acesso à estrutura de componentes monitorados através de *combo lists* como mais um método alternativo;
- Disponibilização de estatísticas, de gráficos pré-definidos e de informação global para todos os componentes da estrutura monitorada, desde o *cluster* à máquina;
- Acesso a componentes de determinada máquina e construção de gráficos de monitorização com base na associação de métricas de diferentes componentes (desde que compatíveis na natureza/tipo de dados);
- Pesquisa de componentes;
- Seleção temporal dos dados, por exemplo, optar por obter os resultados das últimas duas horas ou do dia anterior;
- Comparação direta de dois componentes, representando gráficos com as métricas partilhadas;
- Construção de *dashboards* personalizados com seleção de gráficos.

Partindo da identificação realizada dos problemas mais comuns em clientes web de monitorização, mas também neste conjunto de características desejáveis, foram identificados requisitos que se consideram de valor acrescentado a um sistema de monitorização, por forma a facilitar a concretização dos seus objetivos:

- Oferecer diferentes vistas da estrutura do sistema, facilitando uma mais rápida identificação dos componentes. Dentro das possibilidades, destacam-se a necessidade de vistas que ofereçam:
 - Representação visual hierárquica da estrutura, dos *clusters* até às métricas analisadas;

Descrição do problema

- Perceção de partilha de recursos, de forma a concluir que componentes que executam funções semelhantes;
 - Análise da dimensão do sistema;
 - Representação lógica do sistema, onde a pesquisa é feita não através dos componentes físicos, mas dos serviços/responsabilidades do sistema, navegando até às máquinas que os analisam.
- Rápida construção de gráficos, através da seleção de métricas numa das quatro vistas anteriores, e fácil inclusão em *dashboards*;
 - Fácil e ampla personalização de todos os elementos do ambiente.

No entanto, a solução não deve focar-se apenas aspetos funcionais. Existem requisitos não funcionais essenciais para que uma solução esteja protegida contra falhas e ataques. Pretende-se uma proposta:

- Desenvolvida sob um ambiente que mantenha elevados níveis de performance, para que a utilização em escala não fique comprometida;
- Fiável, para que não exista perdas de informação que possam comprometer a análise correta do sistema;
- Disponível em qualquer dispositivo com acesso à internet;
- Segura, para que a informação contida não seja exposta de forma a comprometer o sistema;
- Intuitiva e de simples utilização, para que não existam perdas de tempo desnecessárias em situações de alerta.

Resultante da análise de requisitos, foram idealizados todos os casos de uso que contribuem para uma solução completa. O anexo A apresenta de forma tabelada os casos de uso, com informação acerca do papel do ator, descrição da funcionalidade, prioridade de implementação (1 - 5, em que 1 significa pouco prioritário e 5 muito prioritário) e pré condições para que a implementação seja possível.

3.3 Sumário

O presente capítulo identifica os principais problemas encontrados nos sistemas analisados no capítulo anterior. Define o foco do tema de estudo e apresenta os tópicos onde a solução proposta irá incidir de modo a melhorar os métodos de análise e extração de informação útil em sistemas de monitorização. Conclui-se que existem aspetos relevantes e que devem fazer parte da solução proposta. No entanto, existe uma clara oportunidade de melhoria da representação gráfica dos sistemas, tornando-os mais objetivos.

O capítulo seguinte especifica a solução proposta, abordando as tecnologias escolhidas e como interagem entre si e o fluxo de utilização lógica da solução, auxiliada por *mockups*.

Capítulo 4

Especificação da solução proposta

O presente capítulo faz referência à implementação da solução proposta. Dado que a complexidade inerente ao volume e heterogeneidade dos dados de monitorização pode ser um entrave à descoberta de problemas e da sua origem, como referido, procurou uma solução fácil de utilizar, com as funcionalidades localizadas em uma só página, para que o acesso a todos os componentes seja também rápido.

As próximas secções descrevem o estudo da interface de utilização e o normal fluxo de utilização. São explicadas as implementações dos quatro principais componentes presentes no sistema: Representação da estrutura da rede, disponibilizada sob cinco vistas diferentes; Janela de seleção de componentes; Todos os tipos de gráficos e suas personalizações; *Dashboards*.

4.1 Estudo da interface

Resultante do estudo dos requisitos, foi desenhada uma interface, em forma de *mockups*, como ilustração da proposta apresentada.

Especificação da solução proposta

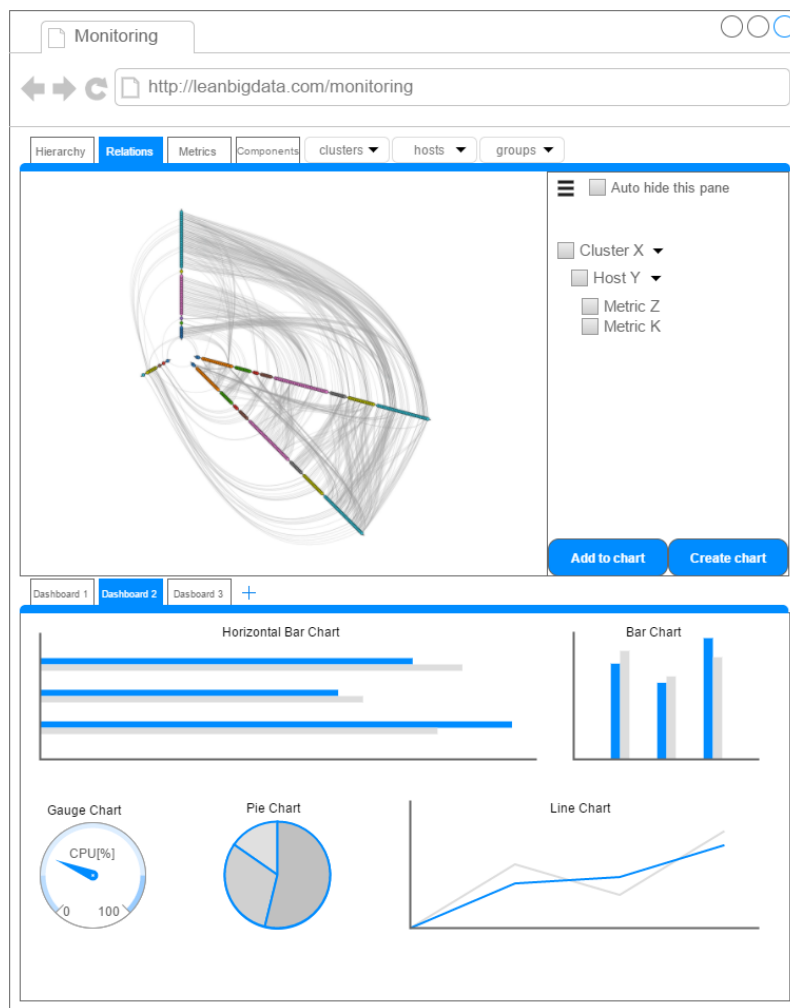


Figura 4.1: Desenho do aspeto visual da solução

Como se pode verificar na figura 4.1, a solução proposta é instanciada numa aplicação *single page*, com os principais elementos da interface facilmente acessíveis, sem necessidade de navegação e descoberta ao longo da aplicação. No topo existem várias hipóteses de representação da estrutura da rede, de forma a oferecer ao utilizador de vários formatos visuais que o auxiliem na pesquisa de componentes do sistema. Essas representações devem ser capazes de relacionar componentes, por forma a facilitar a perceção da organização da rede. Na figura 4.1, ao topo e à direita é também apresentada uma vista de seleção que mostra os componentes selecionados nas vistas. Esta vista suporta também a seleção de métricas, por componente.

Na parte inferior da figura 4.1 está representada a lista de *dashboards*. De salientar que, através da vista de seleção, o utilizador pode selecionar métricas e adicioná-las a um gráfico novo ou a um gráfico existente no *dashboard* presentemente ativo. As várias opções de visualização permitem que o utilizador possa usufruir de várias alternativas para rápida localização dos componentes. Cada uma das vistas de representação da estrutura da rede tem um objetivo de utilização diferente, no entanto estas vistas complementam-se.

4.2 Diagrama de fluxo

Com base na proposta de interface apresentada, a figura 4.2 representa o fluxo de utilização natural na construção de um *dashboard* pelo utilizador.

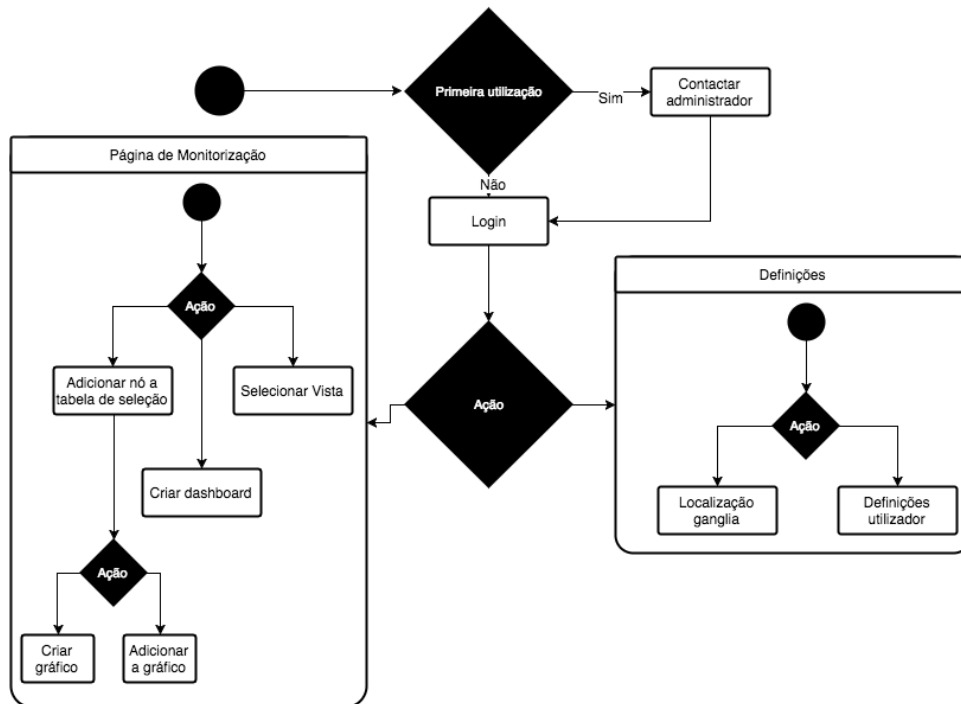


Figura 4.2: Diagrama do normal fluxo de uso

Por se tratar de um sistema com acesso controlado, o registo de novos utilizadores não está acessível publicamente, sendo necessária a intervenção de um administrador do sistema. Após o utilizador entrar na aplicação com as credenciais fornecidas por um administrador é possível aceder à monitorização, onde existem três ações possíveis: seleção da vista global da estrutura, seleção/adição de um componente na vista de seleção ou ainda criação de um *dashboard*. Caso o utilizador opte pela segunda alternativa, pode proceder à criação de um novo gráfico ou à inserção dos componentes selecionados a um gráfico já existente no *dashboard* selecionado.

4.3 Representação visual do sistema monitorado

A oferta de várias vistas alternativas mas complementares de representação da estrutura monitorada ajuda o utilizador a perceber melhor quais as relações que existem entre todos os componentes do sistema monitorado. Como um sistema de monitorização sugere, a rápida identificação de elementos do sistema e a noção da forma como este está disposto, são fatores essenciais para a deteção da origem de problemas e alteração das tarefas que cada componente realiza.

Especificação da solução proposta

A proposta de solução apresentada oferece variadas interações entre o utilizador e cada vista, sendo que existem interações comuns a todas as vistas e outras que são específicas de cada uma. Pertencem ao grupo de interações comuns a todas as vistas, as possibilidades de:

- Selecionar componentes. Esta seleção reflete em uma adição do componente a uma vista de componentes selecionados e alteração do estado desse componente em todas as vistas, alterando a sua cor, por forma a identifica-lo como selecionado;
- Efetuar um *reset* aos componentes selecionados, retirando da janela de seleção todos os componentes adicionados nas vistas de representação da estrutura;
- Realizar técnicas de aproximação e movimentação do gráfico apresentado na vista, facilitando a navegação do utilizador na identificação de componentes

As seguintes secções descrevem cada uma das vistas alternativas do sistema, destacando as suas principais aplicabilidades.

4.3.1 Representação hierárquica

Normalmente, um sistema distribuído encontra-se dividido por *grids* que são compostas por *clusters*. Uma grid é caracterizada por um conjunto de computadores situados, ou não, no mesmo espaço físico que trabalham para um mesmo objetivo. Já um *cluster* é definido como um aglomerado de máquinas conectadas e que partilham um mesmo espaço físico. Cada uma dessas máquinas pode conter elementos de monitorização em recursos críticos como o CPU, a memória, I/O ou a persistência.

Ao nível da representação, a estrutura acima descrita beneficia de elevada compatibilidade com uma estrutura hierárquica de árvore e que pode ser representada através do gráfico presente na figura 4.3.

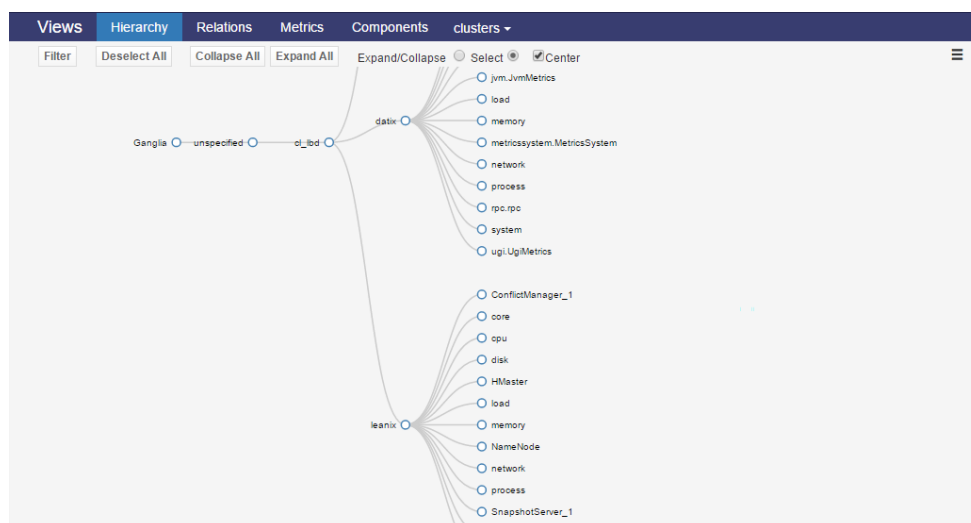


Figura 4.3: Vista de representação hierárquica

Especificação da solução proposta

A representação observada na figura 4.3 ilustra a estrutura do sistema de testes, monitorado em profundidade. No entanto, de forma a evitar problemas de complexidade, o último nível desta estrutura é relativo à família das métricas analisadas, ao invés das métricas em si. No exemplo da figura 4.3 pode-se observar que a máquina com o componente "disk", descendente da máquina "datix" não possui representadas as métricas referentes a esse grupo. Adicionalmente às técnicas de interação comuns a todas as vistas, esta representação possibilita ainda as seguintes funcionalidades, presentes no canto superior à esquerda da vista:

- A seleção de um componente representado na estrutura com um clique do rato resulta na adição à janela de seleção ou no colapsar/expandir o nó clicado, dependendo do modo selecionado ("expand/colapse" ou "select");
- Centragem da representação na janela em função do elemento selecionado;
- Colapsar a representação gráfica da estrutura no seu nó raiz ou expandir todos os nós, por forma a apresentar todos os componentes e famílias de métricas.

4.3.2 Representação relacional

Conjuntamente com a noção de profundidade obtida com a representação hierárquica, considera-se relevante disponibilizar uma representação capaz de apresentar as relações existentes componentes do sistema monitorado e famílias de métricas, isto é, permitir aferir que componentes do sistema estão a analisar mesmas métricas. No caso de métricas sobre funções lógicas, permite perceber que máquinas estão a realizar tarefas semelhantes, oferecendo conhecimento para o desenho de uma nova solução de distribuição de tarefas. É com este propósito que a representação relacional, ilustrada na figura 4.4, foi incluída na solução.

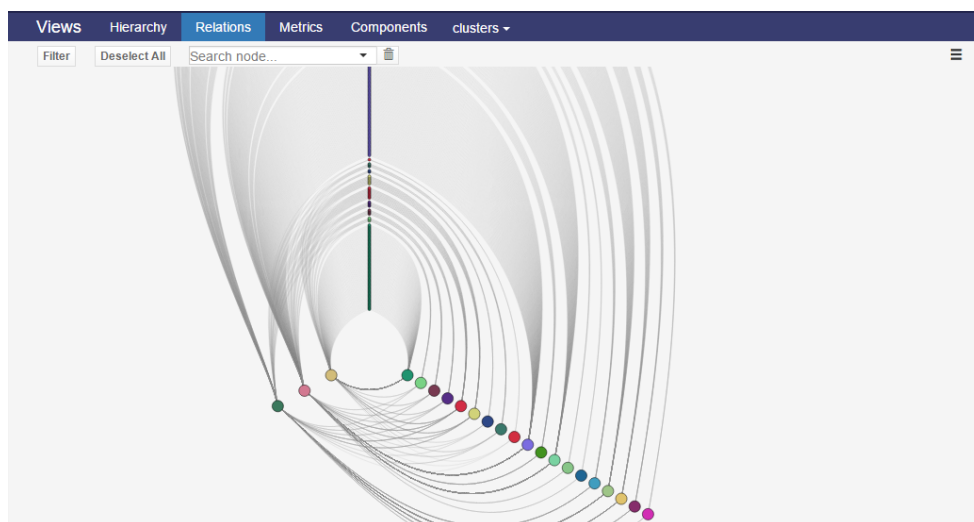


Figura 4.4: Vista de representação relacional

Especificação da solução proposta

Esta representação é apresentada em três eixos onde cada eixo apresenta nós que representam os componentes do sistema, as máquinas físicas e as famílias de métricas. A unir estes nós encontra-se a representação de relações.

No eixo das abcissas, estão representadas as máquinas da rede. Neste caso não existe divisão de máquinas por *cluster*, todas as máquinas da rede são representadas da mesma forma. O eixo das ordenadas apresenta todas as famílias de métricas analisadas. Finalmente, no eixo das cotas são visíveis todas as métricas analisadas no sistema e que decompõem as famílias do eixo das ordenadas. As ligações visíveis entre os componentes estabelecem relações de posse. Ou seja, uma ligação entre um componente do eixo dos yy com o eixo dos xx significa que aquela família de métricas é analisada na máquina correspondente.

Nesta representação cada nó e ligação possuem uma coloração própria. Por forma a facilitar a interpretação, a cor das métricas apresentadas no eixo das cotas é igual à cor correspondente da sua família no eixo das ordenadas. Essa cor apenas é alterada, à semelhança do que acontece em todas as vistas, quando o componente se encontra selecionado.

A representação relacional deve possuir interações próprias, permitindo ao utilizador seleccionar nós e ligações. As consequências de cada uma destas ações são as seguintes:

- Clique em nó no eixo das abcissas: são adicionadas à janela de seleção todas as métricas analisadas na máquina selecionada;
- Clique em nó no eixo das ordenadas: são adicionadas todas as métricas que compõem a família selecionada, analisadas em cada uma das máquinas da rede;
- Clique em nó no eixo das cotas: correspondendo a uma métrica apenas, neste caso é adicionada à janela de seleção de métricas analisadas em cada máquina da rede;
- Clique em uma ligação entre componentes das abcissas e ordenadas: neste caso, a ligação refere a posse de métricas de análise da família do eixo das ordenadas na máquina do eixo das abcissas. São adicionadas à vista de componentes selecionados as métricas que compõem essa família, mas apenas analisadas na máquina referida;
- Clique em uma ligação entre componentes das ordenadas e das cotas: a ligação refere que a métrica representada no eixo das cotas pertence à família do nó das ordenadas. Não existindo nenhuma máquina nesta relação, o clique resulta na adição da métrica presente na ligação e que é analisada em cada uma das máquinas da rede;
- Clique em uma ligação entre componentes das cotas e das abcissas: a ligação representa a análise da métrica no eixo das cotas, na máquina representada pelo nó do eixo das abcissas. Neste caso, é adicionada apenas a métrica analisada nessa máquina.

Devido à complexidade que esta representação pode apresentar o nome de cada um dos componentes está oculto. Uma solução para a identificação pode ser obtida através de um campo de pesquisa, onde o utilizador pode procurar qualquer componente presente no gráfico, dando-lhe visualmente destaque através dos atributos cor e dimensão.

4.3.3 Representação dimensional

Um dos objetivos da análise do sistema monitorado é perceber a dimensão da própria função de monitorização. Por exemplo perceber que máquina tem o maior número de métricas em análise. Este conhecimento é também um fator essencial na perceção do sistema. Na figura 4.5 está representada a solução proposta para a resolução desta problemática.

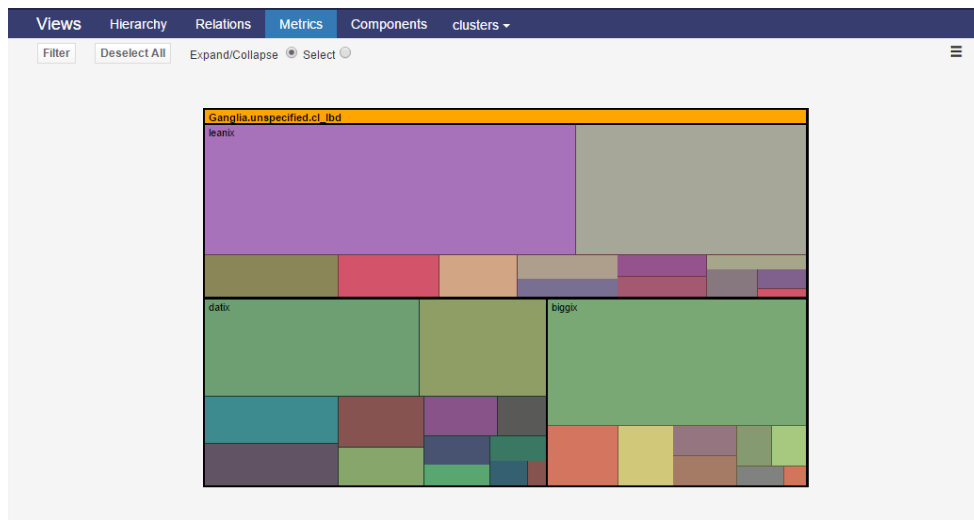


Figura 4.5: Vista de representação dimensional

A solução proposta usa uma representação do tipo 4.5 que é composta por uma figura retangular dividida em várias áreas, simulando a dimensão de cada objeto. No caso concreto da solução aplicada a um sistema de monitorização, é observado uma variação no conteúdo do 4.5, decorrente da navegação através desta vista. Esta solução oferece métodos de pesquisa em profundidade, disponibilizando métodos de aproximação de componentes através da sua seleção. A solução sugere duas possíveis ações decorrentes do clique em um componente da vista, representadas no formulário em cima ao lado esquerdo:

- Expandir/colapsar: o clique de um componente resulta em uma modificação do conteúdo do gráfico. Caso seja clicado na barra cabeçalho, o conteúdo do gráfico muda para o nível de profundidade imediatamente antes. Se pelo contrário, for clicado um componente do gráfico, o seu conteúdo é modificado pela informação desse componente;
- Selecionar: o clique de um componente resulta na adição desse componente, com os seus respetivos filhos, à janela de seleção de componentes.

4.3.4 Representação por listas

A solução proposta pretende manter a capacidade de se realizar uma seleção simplificada e mais aproximada ao que acontece atualmente em outras ferramentas. No entanto, uma vez que a sua utilização ficaria muito próxima do que é possível com a solução hierárquica ou até com a

solução de vista dimensional, esta solução opta por abordar a seleção de forma inversa. Ou seja, a seleção simplificada é composta por três listas sequenciais de elementos que contêm os *clusters*, as máquinas e famílias de métricas. Neste processo de seleção uma lista só fica visível após a seleção de um item na lista imediatamente anterior: inicialmente apenas a lista de *clusters* está visível e apresenta todos os *clusters* do sistema, não selecionados à partida. Um vez selecionado um *cluster*, são adicionadas à janela de seleção todas métricas analisadas nas máquinas que compõe esse *cluster*. Consequente da seleção de apenas um *clusters* na primeira lista, a lista seguinte fica visível com todas essas máquinas selecionadas. O conceito de abordagem inversa está presente nesta funcionalidade. Neste caso, uma vez que todos os componentes se encontrado selecionados, o utilizador pode remover componentes de modo a ficar apenas um selecionado e a lista seguinte fique desta feita, também visível. A figura 4.6 ilustra um exemplo em que a primeira e segunda listas apenas contêm um componente selecionado, permitindo que a lista seguinte esteja visível. Para o presente caso, as métricas sombreadas a azul estão presentes na vista de componentes selecionados.

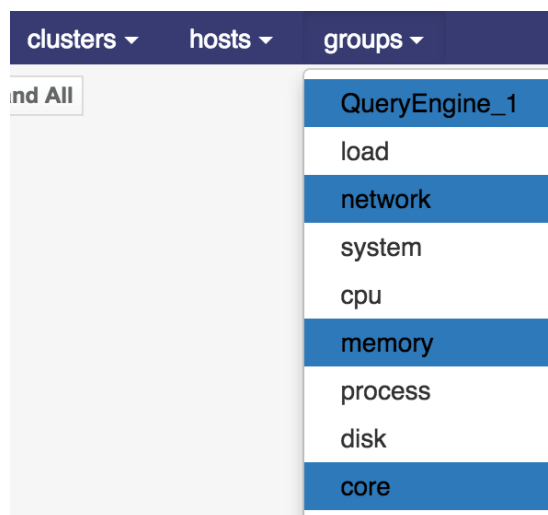


Figura 4.6: Vista de representação por listas

4.3.5 Representação lógica

Uma das representações verdadeiramente diferenciadoras face a outras soluções é a capacidade de representação o sistema monitorado através da perspetiva lógica. Um sistema, da mesma forma que pode ser organizado por hierarquia, também pode ser representado tendo como critério as responsabilidades dos componentes lógicos. Isto é, quando existe num *cluster* duas máquinas que realizam ações semelhantes, estas também podem ser agrupadas. Sendo assim, a figura 4.7 ilustra a representação lógica do sistema monitorado que é, neste caso, o sistema gestor de base de dados LeanXscale.

Especificação da solução proposta

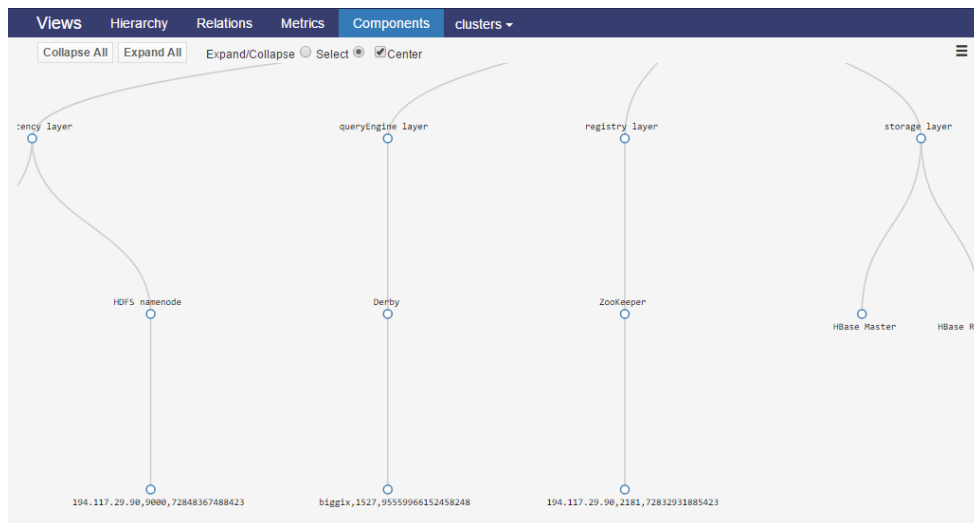


Figura 4.7: Vista de representação lógica

O sistema em causa contém diversos tipos de funções a desempenhar, sejam elas pertencentes a camadas de persistência, de motor de pesquisa ou de registo. Cada uma destas camadas contém diversas aplicações de suporte, como é o exemplo do "Derby" para o "query engine". Todas as aplicações descendem de camadas contendo, como descendentes, as máquinas onde estas desempenham funções.

Desta forma a solução proposta oferece ao utilizador uma representação pouco habitual nos sistemas de monitorização atuais, mas muito prática e funcional. Com a representação lógica da rede facilmente se podem retirar conclusões de sobrecargas, distribuições de tarefas e origem dos problemas.

Esta solução oferece um objetivo um pouco diferente das representações apresentadas anteriormente: ao invés de adicionar componentes à janela de seleção, para posterior demonstração dos valores das métricas, a solução agrupa os valores de métricas, podendo oferecer o valor máximo ou a média dos valores da métrica analisada em todas as máquina.

Esta representação necessita de metadados da estrutura e de agrupamento de métricas por componentes, que podem ser configurados por ficheiros auxiliares. Estes ficheiros contêm informação acerca das camadas do sistema monitorado, que serviços integram cada camada e qual o agrupamento esperado para cada um dos componentes presentes num serviço, podendo estes agrupamento ser a média, soma, valor máximo ou valor mínimo.

Do ponto de vista da implementação da prova de conceito, uma vez que se optou por estes ficheiros estarem presentes no servidor Ganglia, torna-se necessário que a solução comunique com o sistema operativo para obter a informação contida nesses ficheiros, seguindo o seguinte procedimento:

1. Extrair informação de um ficheiro XML principal, que contém informação acerca da organização das camadas na rede, com as respetivas aplicações associadas;

Especificação da solução proposta

2. Com o auxílio da ferramenta Zookeeper, é possível obter quais as máquinas que integram as aplicações obtidas. O Zookeeper é um serviço centralizado de manutenção e configuração de informações acerca dos componentes do sistema. Este serviço aloja informação acerca do nome das máquinas, sincronização de tarefas, serviços, entre outros. Através do uso deste é possível saber quais as máquinas que contêm cada serviço;
3. Obter ainda do ficheiro informação de quais as métricas que são analisadas em cada serviço, complementadas com o agrupamento esperado para representação gráfica e se esse gráfico deve estar ou não visível numa construção automática de um *dashboard* associado a um serviço.

À semelhança da solução apresentada nas outras representações, esta proposta também deve disponibilizar interações que facilitam e proporcionam a mais ágil utilização da informação: o processo de criação de *dashboard* é semelhante ao processo executado nas outras vistas e começa pela seleção de um componente no gráfico. No entanto, apenas podem ser selecionados os componentes presentes nos dois últimos níveis de profundidade (um serviço ou uma máquina), sendo essa seleção exclusiva a apenas um nó. Uma vez selecionado um nó, este é adicionado à janela de componentes selecionados exclusiva desta vista, ilustrado na figura 4.8. Nesta janela são apresentadas as métricas agrupadas conforme a informação presente nos ficheiros de configuração. Cada componente da vista apresenta-se selecionado ou não por defeito, coerente com a informação dos ficheiros. No entanto, antes da criação do *dashboard*, é possível alterar essa seleção pré-definida, de modo a esses componentes integrarem o resultado final. A informação relativa ao agrupamento dos valores de métricas é aplicado quando o nó selecionado na vista for referente a um serviço. Neste caso, cada gráfico do *dashboard* deve representar o valor da métrica agrupada entre todas as máquinas que analisam o serviço, conforme a descrição no ficheiro.

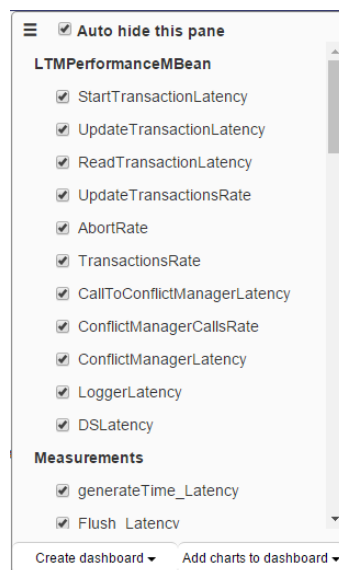


Figura 4.8: Janela de componentes selecionados para a vista lógica

4.4 Filtragem de componentes

Com a projeção de um sistema com cerca de mil máquinas em análise, prevê-se que as vistas de estrutura poderão elevar a sua complexidade e resultar numa leitura débil devido a visual *clutter*. De modo a solucionar este problema, optou-se por introduzir um campo de opção de seleção de *clusters* e máquinas. Neste o utilizador ganha a capacidade de seleccionar os nós que pretende que estejam representados nas vistas de estrutura. Dessa forma, a complexidade será aquela que o utilizador seleccionar, podendo optar por filtrar o seu alvo de pesquisa a poucas máquinas e assim centrar a atenção em componentes específicos, ou então alargar o seu campo de visão e efetuar um filtro mais pequeno, mas que lhe ofereça uma realidade mais aproximada do sistema.

4.5 Janela de componentes seleccionados

A janela de componentes seleccionados está presente em todas as vistas de representação da estrutura do sistema monitorado. A seleção de um componente numa das diferentes vistas disponibilizadas deve resultar na adição do componente à janela de componentes seleccionados. No entanto, esta janela apenas mantém coerência de seleção entre as vistas de hierarquia, dimensão, relação e de lista. Para estes casos, cada componente seleccionado mantém uma coerência de seleção entre as diferentes vistas. A ilustração da figura 4.9 sugere um exemplo do seu estado.

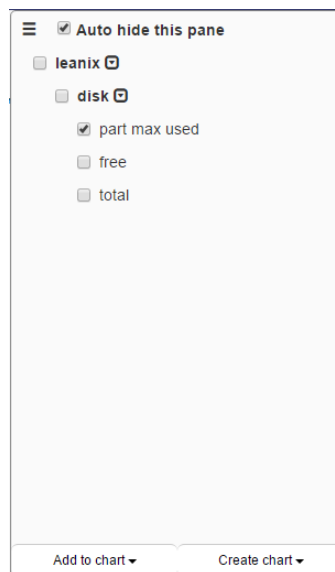


Figura 4.9: Janela de componentes seleccionados

A organização dos componentes presentes na tabela da figura 4.9 apresentam sempre a mesma estrutura, independentemente da vista a partir do qual os componentes são seleccionados. Todos os componentes adicionados são organizados hierarquicamente, agrupados por máquinas e grupos de métricas. A janela oferece a opção de "Auto hide this pane". Esta opção disponibiliza um modo em que a janela apenas fica visível se existir algum componente seleccionado nas vistas.

Uma vez que a presente vista contém elementos, estes também são passíveis de ser seleccionados. Esta seleção pode ser feita em camadas superiores, de forma a afetar recursivamente todos os descendentes. Após a seleção feita podem ser criados os diversos tipos de gráficos, a ser adicionados ao um *dashboard* ativo no momento, ou então adicionar os componentes seleccionados a gráficos já existentes no *dashboard* ativo.

Paralelamente a esta janela existe uma outra, que a substitui no caso da vista de componentes. Neste caso, a organização hierárquica faz pouco sentido, uma vez que estaria contra o principal objetivo desta vista, a sua organização lógica e não física. A figura 4.8 presente na secção 4.3.5 ilustra uma situação neste caso.

A organização das métricas na janela é feita através da informação recolhida pelos ficheiros de configuração, que agrupam as métricas por tipos de análise. A seleção deste nós tem um valor por defeito, também declarado nos ficheiros de configuração, podendo ser alterada antes da criação de um *dashboard*. O *dashboard* resultante será constituído por um gráfico para componente seleccionado. A solução proposta oferece diferentes tipos de gráficos que podem integrar os *dashboards*. A par da opção de criação de *dashboard* existe a opção de criação de novos gráficos, onde são adicionados os gráficos das métricas seleccionadas ao *dashboard* ativo.

4.6 Tipos de gráfico de monitorização

Apesar dos gráficos de linhas serem considerados dos melhores para representação de dados lineares, existem outras soluções de representação de dados de monitorização, com diferentes características, capazes de representar valores de desempenho, podendo ser usados mediante diferentes objetivos. A construção de um gráfico de monitorização inicia-se com a seleção de um tipo de gráfico (ver figura 4.10), disponível a partir da vista de componentes seleccionados, depois de se seleccionar uma ou mais métricas.

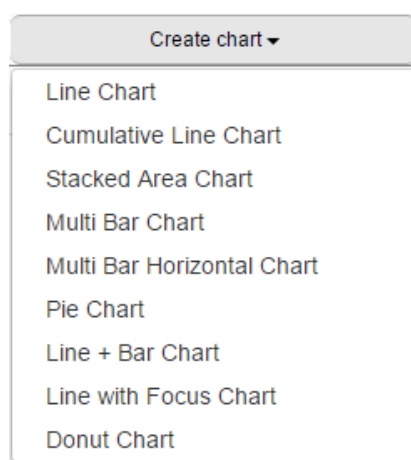


Figura 4.10: Lista de gráficos disponíveis

Especificação da solução proposta

A proposta de solução considera a criação de gráficos do lado do cliente, sendo este processo suportado na biblioteca NVD3, com a qual foram realizados estudos de todos os tipos de representação gráfica que esta biblioteca oferece e escolhidos aqueles que se consideram obter os melhores benefícios para um sistema de monitorização. Quando adicionado a um *dashboard*, um gráfico apresenta um cabeçalho representado na figura 4.11. Nesse cabeçalho, do lado esquerdo está presente o título do gráfico e ao lado direito as possibilidades de aceder ao menu de opções (símbolo de roda dentada), parar ou recomeçar a atualização do gráfico (símbolo de "pause" ou "play" respetivamente) e apagar o gráfico (símbolo do caixote do lixo).



Figura 4.11: Exemplo de um cabeçalho de um gráfico

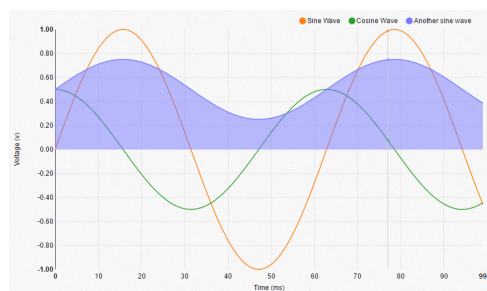
A oferta de uma vasta gama de tipos de gráfico não é suficiente para que as problemáticas fiquem solucionadas. A solução oferece um conjunto de interações, nomeadamente a possibilidade de ocultação de métricas de análise no gráfico. Existem outras interações presentes nos diferentes tipos de gráfico e que são explicadas nas secções seguintes.

As próximas secções apresentam os tipos de gráficos integrantes na solução e de que forma contribuem para que a solução responda aos problemas encontrados.

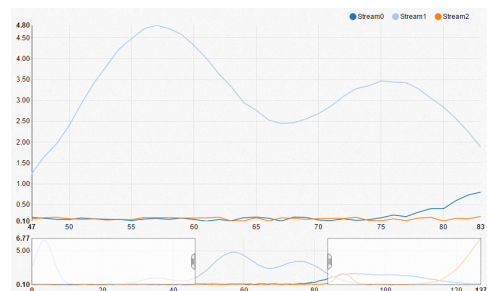
4.6.1 Gráfico de linhas

O gráfico de linhas é o tipo de gráfico normalmente usado para representar funções lineares e temporais. Usufrui do fator de representação em série, sendo útil para comparação de funções com a mesma escala de medida.

A figura 4.12 ilustra duas representações de gráficos de linhas. Com a representação ilustrada na figura 4.12b é permitido ao utilizador definir dois limites temporais para que o gráfico em cima apenas represente os valores compreendidos entre esses limites, facilitando a análise específica de um instante temporal.



(a) Exemplo sem opção de foco



(b) Exemplo com opção de foco

Figura 4.12: Gráficos de linhas

4.6.2 Gráfico de linhas acumulativo

A representação de dados de monitorização através de um gráfico de linhas cumulativo, apesar de visualmente ser muito semelhante ao gráfico de linhas, apresenta uma característica essencial na comparação de valores ao longo do tempo. Esta solução oferece a possibilidade de definir um ponto de comparação, para que todas as funções sejam representadas com base nesse instante. A definição desse ponto de comparação é apresentada com uma linha vertical a vermelho no exemplo da figura 4.13. No que diz respeito à análise de desempenho, esta solução gráfica oferece duas vantagens diferenciadoras:

- Com a representação de duas métricas diferentes analisadas na mesma máquina, esta solução revela-se útil para capacitar a deteção de relação comportamental entre as métricas;
- Com a representação de métricas iguais mas analisadas em diferentes máquinas, beneficia-se da vantagem óbvia de comparação de desempenho entre máquinas.

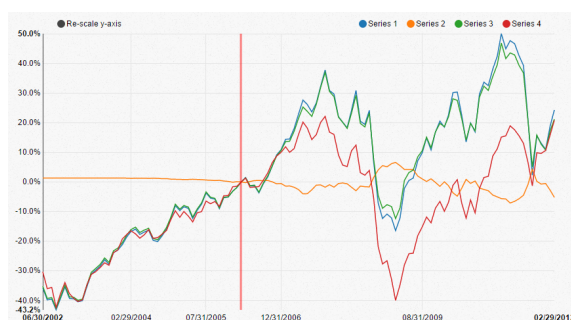


Figura 4.13: Gráfico de linhas acumulativo

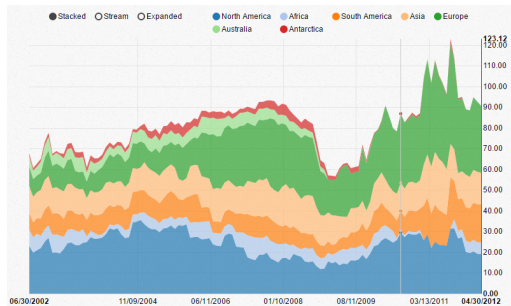
4.6.3 Gráfico de área

A representação de dados de monitorização através de um gráfico de área distingue-se pelo preenchimento da área abaixo de cada função. A representação de cada métrica em diferentes áreas beneficia o objetivo de comparação de métricas no mesmo gráfico. Existem três opções de representação deste tipo de gráfico:

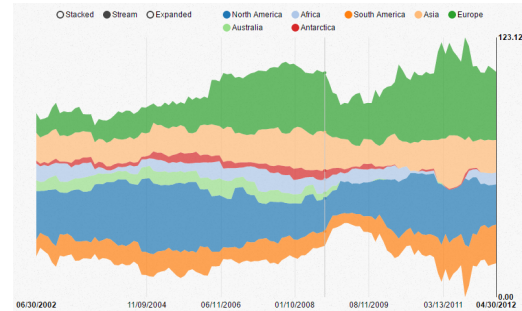
- Stacked (Empilhada): A figura 4.14a ilustra esta opção, onde são disponibilizados os valores na unidade de medida que estes são analisados. Uma vez que estes são empilhados, facilmente é possível retirar conclusões acerca do somatório de todas as funções analisadas;
- Stream: Este modo, presente na figura 4.14b, apresenta uma forma peculiar de representação de funções. Neste método é adicionada a primeira função simetricamente a um eixo linear y imaginário. Todas as funções seguintes são empilhadas de forma a comprimir sucessivamente a sua área;

Especificação da solução proposta

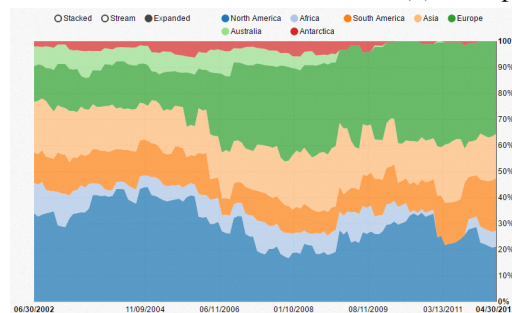
- **Expanded (Expandida):** Neste formato, representado na figura 4.14c, a representação é semelhante ao primeiro caso, no entanto com escala percentual, mantendo sempre o mesmo volume total (100%), variando apenas a taxa de ocupação em cada momento.



(a) Exemplo de área empilhada



(b) Exemplo de área *stream*



(c) Exemplo de área expandida

Figura 4.14: Gráficos de área

4.6.4 Gráfico de barras

A representação de dados de monitorização através de um gráfico de barras oferece uma visualização semelhante à do gráfico de linhas, mas com representação de cada instante através de uma barra sem conexão ao ponto seguinte. Esta solução apresenta vantagens numa representação onde se pretenda a comparação direta de funções em instantes precisos. Para esta representação existem duas opções:

- **Grouped (agrupado):** Ilustrado na figura 4.15a, este gráfico oferece a cada instante os valores das funções representados por uma barra.
- **Stacked (Empilhado):** Com esta opção, ilustrada na figura 4.15b, as funções são representadas para cada instante numa só barra, empilhando os valores de todas as funções.

Especificação da solução proposta

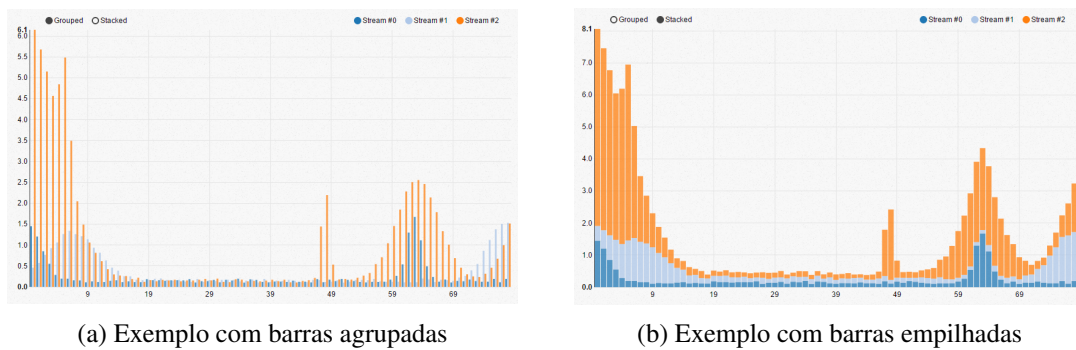


Figura 4.15: Gráficos de barras

4.6.5 Gráfico de linhas e barras

A representação de dados de monitorização através de um gráfico que integra linhas e barras reúne as vantagens dos gráficos constituintes. A necessidade da disponibilização deste no conjunto de soluções surge do facto de por vezes duas funções no mesmo gráfico terem ordens de grandeza muito diferentes. Como ilustrado na figura 4.16, estão presentes duas escalas distintas, através de dois eixos de ordenadas.

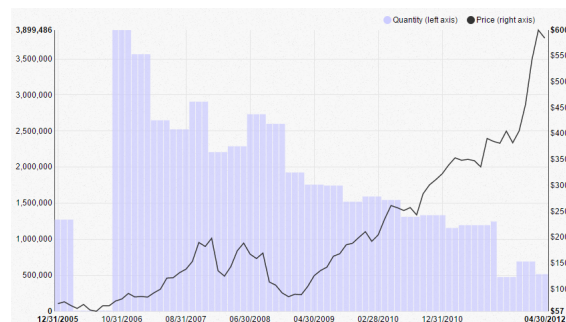


Figura 4.16: Gráfico de linhas e barras

4.6.6 Gráfico de barras horizontais

Ao contrário das soluções apresentadas anteriormente, a representação de dados de monitorização através de um gráfico de barras horizontais não reproduz dados de uma série temporal. Cada um dos grupos é identificado como uma métrica de análise, onde os valores de desempenho representam valores obtidos num instante particular. Desta forma, torna-se simplificada a tarefa de comparação de estado dos componentes analisados em tempo real, sendo possível observar o último valor de análise recolhido.

Especificação da solução proposta

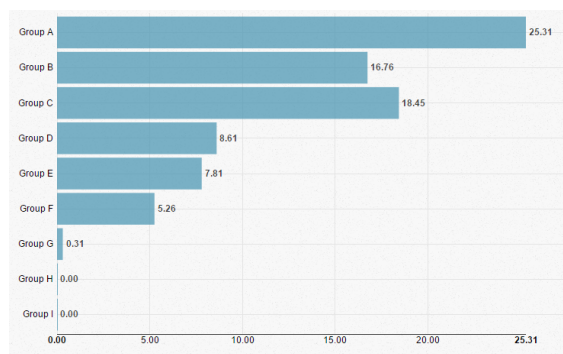


Figura 4.17: Gráfico de barras horizontais

4.6.7 Gráfico circular

A representação de dados de monitorização através de um gráfico circular é útil quando se pretende mostrar apenas um valor por métrica, usualmente o último recolhido. Este gráfico oferece a possibilidade de representar os dados em formato percentual, tornando-se muito útil quando o utilizador pretende obter comparações de métricas semelhantes.

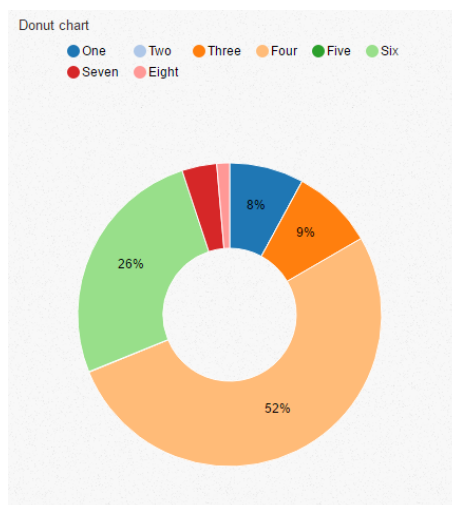


Figura 4.18: Gráfico circular

4.6.8 Personalização gráfica

A análise rápida de dados de monitorização a partir de *dashboards* preenchidos com muitos gráficos pode revelar-se uma tarefa complicada. Por este facto, a personalização gráfica é considerada um aspeto fundamental, de modo a possibilitar criar elementos visuais diferenciadores. Assim, considera-se indispensável que cada gráfico tenha associado um conjunto de personalizações, acessíveis a partir do menu de opções, ilustrado na figura 4.19.

Especificação da solução proposta

The image shows a 'Chart Settings' window with the following fields and options:

- Name:** Example Chart
- Type:** Line Chart (dropdown menu)
- Update Rate (s):** 15
- From:** [date picker]
- To:** [date picker] now ☒
- X Label:** X Label
- Y Label:** Y Label
- Data Series List:**
 - cl_ibd.datix.cpu_idle.sum (selected)
- Series Details (for selected series):**
 - Label:** cl_ibd.datix.cpu_idle.sum
 - Color:** blue
- Buttons:** Apply, Delete

Figura 4.19: Menu de opções de um gráfico

O formulário possibilita a modificação das propriedades seguintes:

- **Nome do gráfico:** Após a criação de um novo gráfico, o sistema atribui automaticamente um nome, com base nos seus componentes. Esse nome irá identificar o gráfico e ficará visível na representação do gráfico no *dashboard*;
- **Tipo do gráfico:** À semelhança do que acontece na criação dos gráficos, o sistema oferece a possibilidade de troca após a sua criação;
- **Período de atualização:** O tempo de atualização dos valores recolhidos é definido em 15 segundos, mínimo possível com o uso do Ganglia. Este valor poder ser configurado por forma a reproduzir atualizações mais lentas;
- **Seleção temporal:** A solução usufrui ainda da excelente capacidade de não só visualizar os últimos valores lidos (por defeito os últimos 15 minutos), mas também os valores que ocorreram em um espaço temporal anterior. O formulário inclui funcionalidades que permitem definir qual os limites temporais para os dados representados;
- **Nomes dos eixos:** Cada gráfico pode conter as mais variadas métricas em análise. Com isto, o eixo das ordenadas pode representar diferentes unidades de medida. A personalização do nome destes eixos ajuda o utilizador a diferenciar melhor estes casos;
- **Funções:** Como foi referido em 3.2, todos os gráficos são compostos por funções. Propriedades como nome e cor são importantes de serem editáveis. Através deste menu é também possível proceder à remoção de funções presentes no gráfico.

Todas as alterações efetuadas no menu têm aplicação imediata no respetivo gráfico, para que o utilizador consiga ver o resultado em tempo real. No entanto, as alterações só ficam guardadas depois de serem gravadas.

Especificação da solução proposta

Considera-se também pertinente oferecer uma representação alternativa quando os gráficos são criados a partir da vista lógica. Esta necessidade nasce do facto dos gráficos criados nesta situação apenas apresentarem uma métrica de análise. Assim, a sua interface apresenta um valor de leitura em grande plano, destacando o ultimo valor recolhido, normalmente o de maior importância para o utilizador. A figura 4.20 ilustra a situação descrita. O utilizador tem sempre a possibilidade de inverter o foco, colocando o gráfico em destaque.

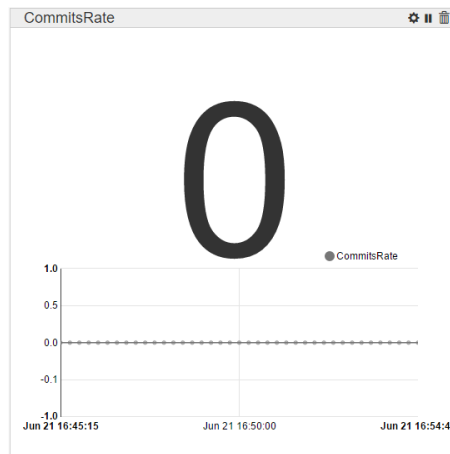


Figura 4.20: Gráfico criado através da vista lógica

4.7 Dashboards

Os *dashboards* revelam-se dos elementos mais importantes num sistema de monitorização, ocupando grande parte do ciclo de vida destes sistemas e sobre o qual o utilizador despende mais tempo de observação. Por conseguinte, considera-se ser importante que um sistema de monitorização ofereça boas capacidades de manipulação e gestão de *dashboards*. O aspeto visual dos *dashboards* da solução proposta é ilustrado na figura 4.21.



Figura 4.21: Vista exemplar de um Dashboard

Na solução proposta pretende-se que *dashboard* seja composto por um conjunto de gráficos, ajustáveis em tamanho e posição (sendo que a personalizações dos gráficos foi prevista e descrita na secção 4.6.8). Na solução proposta, cada *dashboard* apresenta uma tabela virtual de 6 colunas de largura, podendo cada gráfico obter as correspondentes dimensões. A sua movimentação segue o mesmo princípio, sendo que cada elemento tem de estar posicionado no início de cada coluna e ter dimensões inteiras. Na figura 4.21, no topo e lado esquerdo, é possível observar a lista de *dashboards* disponíveis para o utilizador. A adição de novos gráficos através da vista de componentes seleccionados tem efeito no *dashboard* ativo, e será adicionado na posição (0,0) com a dimensão quadrada de duas casas. Ao lado direito é possível para a atualização de todos os gráficos do *dashboard* e aceder ao menu de opções, ilustrado na figura 4.22.

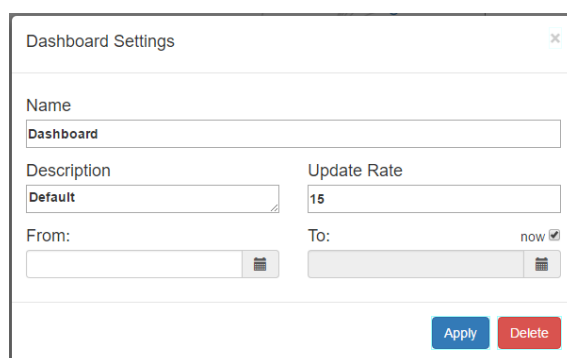


Figura 4.22: Menu de opções dos *dashboards*

As ações possíveis são muito semelhantes às personalizações dos gráficos, sendo que as alterações são aplicadas a todos os gráficos do *dashboard*. Exceccionalmente, existe a opção de descrição, que apenas é visível neste menu, mas pode ser usada para introduzir qualquer descrição que o utilizador considere relevante acerca do elemento.

4.8 Sumário

O presente capítulo detalha os pormenores da solução proposta, baseada na sua interface gráfica e fluxo de utilização. Neste capítulo são propostas representações visuais alternativas e complementares da estrutura do sistema monitorado. Para cada representação detalharam-se as interações que se consideram permitir ao utilizador adicionar os componentes a uma vista de componentes seleccionados auxiliar que deverá ser usada posteriormente para criação de gráficos e *dashboards*. Esta oferta verifica-se ampla e com possibilidade de análises sob diferentes pontos de vista, possibilitando a utilização mais rápida mediante as possíveis intenções.

Com a apresentação da solução para a representação da estrutura da rede, foram ainda clarificados os detalhes das personalizações presentes nos gráficos e *dashboards*.

Este capítulo inclui considerações sobre os tipos de gráficos escolhidos para que a oferta de representação dos dados de monitorização seja a mais completa possível. As soluções apresentadas,

Especificação da solução proposta

bem como todas as possíveis personalizações, contribuem para a criação de aspetos diferenciados entre os elementos, possibilitando a criação de *dashboards* dinâmicos.

O capítulo seguinte apresenta os resultados obtidos com a implementação da solução proposta, realçando os pontos de avaliação a que a solução foi imposta e a concretização de uma resposta ao problema encontrado.

Especificação da solução proposta

Capítulo 5

Implementação da solução

O presente capítulo descreve o desenho da proposta de solução, focando os principais componentes do sistema e a forma como interagem entre si. Esta etapa revela-se importante para a prossecução de uma solução que contenha as funcionalidades identificadas no capítulo anterior, assentes em uma arquitetura sólida, capaz de albergar as camadas de desenvolvimento da solução.

Nas próximas secções são realçados os testes primordiais realizados, para que seja possível instanciar a comunicação entre os diferentes componentes do sistema proposto, nomeadamente entre o Ganglia e a interface com o utilizador.

Posteriormente é apresentado o desenho da arquitetura lógica e modelo de dados do sistema, seguida da pilha tecnológica escolhida para o desenvolvimento da solução proposta. Por fim são realçados alguns detalhes da implementação modular dos componentes de gráficos e *dashboards*.

5.1 Avaliação do modo de comunicação Ganglia

Para realizar o estudo da melhor abordagem à comunicação entre o Ganglia e a interface cliente, foi construída uma rede virtual composta por três máquinas monitorizadas pelo Ganglia. Deste modo procurou-se que este ambiente de avaliação fosse o mais próximo possível de um ambiente de produção.

Para a implementação desta rede foi usada a ferramenta Vagrant, normalmente usada na construção de ambientes virtuais de apoio ao desenvolvimento. A figura 5.1 ilustra o diagrama de virtualização pretendido.

Esta virtualização é constituída por 3 máquinas, identificadas como "Host 1", "Host 2" e "Host 3". Todos os valores recolhidos por cada instância do gmond são comunicados através de *multicast* ou *unicast* para o servidor de gmetad, através da porta 8649. O "Host 1" é responsável pelo armazenamento dos valores em ficheiros RRD e pelo alojamento da aplicação web do Ganglia. Os valores de monitorização podem ser obtidos pelo "Host 1" através da comunicação direta com um serviço presente no Ganglia que realiza a interpretação dos ficheiros RRD para XML, localizado

Implementação da solução

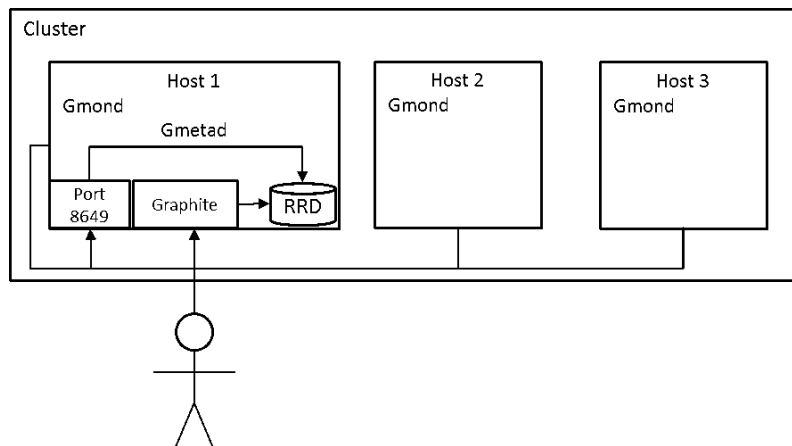


Figura 5.1: Diagrama de representação da arquitetura de testes

na porta 8648. Existe ainda a possibilidade de extração dos valores presentes nos ficheiros através do auxílio da ferramenta Graphite. Nesta etapa, os valores são observáveis através da aplicação cliente do Ganglia instanciada nessa máquina ou através de uma API pública.

Após efetuada uma avaliação ao funcionamento do Ganglia, verificou-se que este usufrui de uma API disponível através da componente da aplicação web. Consequentemente, apoiado pelo facto da aplicação web ser substituída, foram realizados testes de comunicação direta com a máquina que alojava a informação das métricas, o "Host 1". Foram realizadas tentativas de comunicação através de *socket*, com suporte da aplicação RRDTOOL e com auxílio do Graphite. As secções seguintes abordam cada uma das opções analisadas, realçando as suas principais vantagens.

5.1.1 Comunicação baseada em *socket*

Apoiado no facto de existir uma comunicação possível com a porta 8648 da máquina que aloja o serviço do gmetad, responsável pela interpretação dos dados dos ficheiros RRD, foram realizados alguns testes na tentativa de obter conhecimento acerca da informação presente nos ficheiros. Foi concluído que existe a possibilidade de comunicação, sendo possível obter informação acerca dos últimos valores medidos recolhidos. Com isto, a comunicação através de *sockets* foi considerada uma das possibilidades de solução. Esta solução, a ser a escolhida, coloca de parte o armazenamento dos valores de métricas recolhidas em ficheiros RRD e permite passar a utilizar uma base de dados externa. Contudo, é necessária a presença de uma rotina que comunique através da porta 8648 e que armazene os valores recolhidos, por forma a ser utilizada pela aplicação de interface.

5.1.2 Comunicação suportada por RRDTOOL

Considerando a possibilidade da solução proposta ficar alojada no mesmo servidor onde reside a instância de gmetad, foi também estudada a possibilidade de obter os dados diretamente a partir de processamento efetuado sobre os ficheiros RRD. Para isso é necessário conhecer a localização

destes ficheiros e executar a ferramenta RRDTOOL através da *shell* do sistema operativo, capaz de converter os dados presentes nos ficheiros RRD.

5.1.3 Comunicação suportada em Graphite

Analisado o código da aplicação web do Ganglia, constatou-se que esta aplicação comunica com uma instância da aplicação Graphite, instalada na mesma máquina, para obter os dados das métricas em formato gráfico. Depois de analisada a ferramenta (resumida no capítulo do estado da arte), constatou-se ser possível comunicar com esta (por HTTP) e obter os dados em vários formatos.

5.1.4 Sumário

A análise das diferentes opções apresentadas nas secções anteriores permitiu concluir que a inclusão do Graphite como ferramenta intermédia entre a recolha e persistência de dados de monitorização e um cliente web é a opção mais favorável, uma vez que pode ser usada através de um ambiente externo de comunicação (uma API), onde é possível questionar valores de métricas em diferentes momentos.

No entanto, o uso do Graphite implica que a solução proposta tenha de contemplar uma comunicação prévia por *socket*. Essa característica resulta do facto das interrogações a realizar ao Graphite implicarem a referência das métricas específicas. No entanto, o Graphite desconhece a estrutura do sistema monitorado pelo que a comunicação prévia realiza-se com o objetivo da aplicação cliente conhecer a estrutura do sistema. Para isso, deverá ser efetuada uma comunicação *socket*, que devolve toda a estrutura do sistema em formato XML. Uma vez na posse desta informação é possível interrogar o Graphite com a localização precisa das métricas desejadas.

5.2 Arquitetura

Depois de conhecido o melhor método de comunicação com o Ganglia, o estudo da solução providenciou a proposta de arquitetura lógica que se pode observar na figura 5.2.

Nesta figura:

- A camada cliente é composta por toda a estrutura funcional que é executada num *browser*. No cliente está presente a gestão, englobando: todo o processo de configuração do sistema, desde as definições gerais à gestão de contas de utilizador; a estrutura global do sistema, sendo composta pelo conjunto de vistas referentes à estrutura da rede na sua globalidade, representando os *clusters*, máquinas e as suas métricas analisadas; o módulo de gráficos que suporta a gestão dos diferentes tipos de gráficos, gestão de configurações, aparência e identificação; O módulo de *dashboards*, representando a gestão de propriedades e conteúdo de *dashboards*, deste a adição de novos gráficos, à sua organização e dimensão.

Implementação da solução

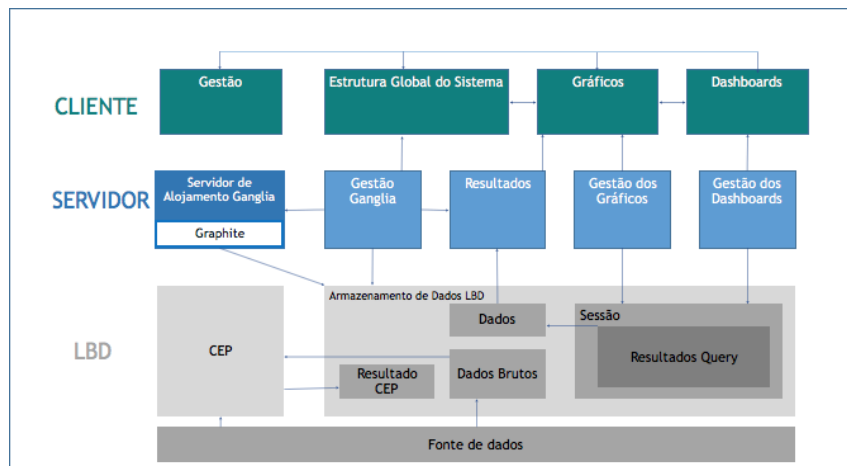


Figura 5.2: Diagrama de arquitetura do sistema desenvolvido

- O grupo do servidor integra as funcionalidades de interação entre o cliente e o sistema gestor de base de dados subjacente, e tem como principal objetivo responder a pedidos do cliente. Neste grupo existem os módulos do servidor Ganglia (e da sua gestão), dos dados de monitorização, da gestão de gráficos e da gestão de *dashboards*. Fazem parte do servidor Ganglia todas as máquinas monitorizadas na rede e a instância do Graphite, que interage com o módulo de gestão do Ganglia. Desta relação obtém-se os dados relativos à estrutura da rede e dos valores das métricas. O módulo de gestão do Ganglia é posteriormente responsável por efetuar o tratamento dos dados e encaminhá-los conforme a sua origem. Caso se tratem de dados de estrutura, reencaminha para o cliente, senão envia para o módulo de resultados do servidor, responsável pela comunicação com os gráficos do cliente. Existem ainda o módulo de gestão de gráficos e de gestão de *dashboards*, responsáveis por responder a pedidos de seleção, inserção, modificação e eliminação de gráficos e de *dashboards*, respetivamente.
- O sistema gestor de base de dados é onde toda a informação e motor de SQL estão alojados. Responsável pelo processamento dos dados, análise, armazenamento e todas as transformações que estes requerem. Corresponde a uma instância do novo SGBD LeanXScale.

5.3 Modelos de dados

Para que todos os casos de uso pudessem ser implementados sem problema, foi necessário construir um modelo de dados que garantisse o seguro e completo funcionamento do sistema. Do modelo de dados, ilustrado na Figura 5.3 destacam-se as tabelas utilizador, gráfico, *dashboard* e *workflow* (sendo que esta última é utilizada por um subsistema que irá integrar a solução proposta).

A tabela utilizador, além das propriedades como nome de utilizador, email e password, pode ter ligações a dois tipos de tabelas: "T_WORKFLOW" e "T_MONITORING_CHART". Esta última é uma tabela descendente da "T_CHART" e contém propriedades como o nome, a descrição, as opções e os dados. A tabela "T_CHART" possui ainda duas ligações:

Implementação da solução

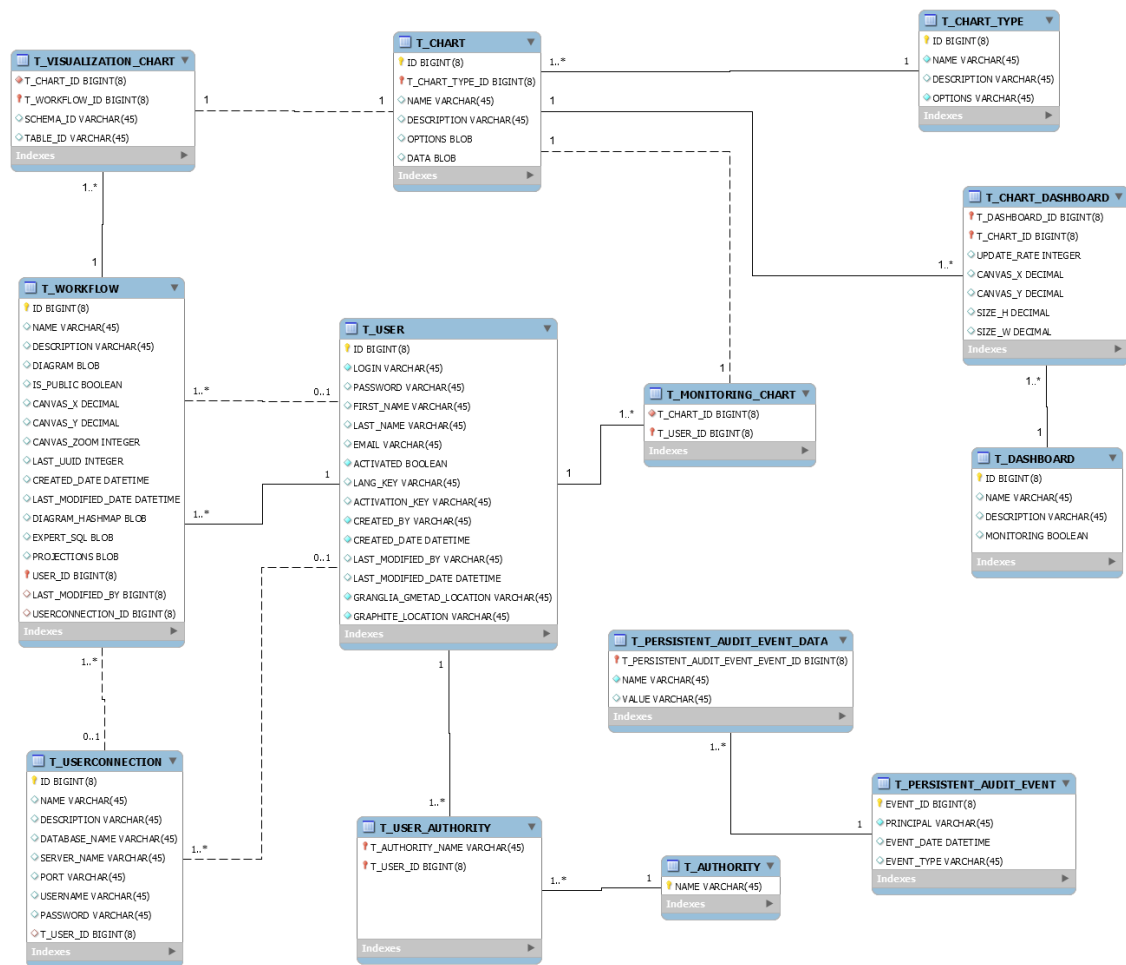


Figura 5.3: Modelo de dados do sistema desenvolvido

- à tabela "T_CHARTTYPE", sendo esta obrigatória e faz referência ao tipo de gráfico;
- à tabela "T_DASHBOARD", contendo uma tabela auxiliar a cada uma das ligações, com o tempo de atualização de dados, posição e dimensões dos gráficos associados ao *dashboard*.

A tabela "T_DASHBOARD", por sua vez, contém um nome e uma descrição. O sistema usufrui ainda de um mecanismo de auditoria, do qual fazem parte as tabelas "T_PERSISTENT_AUDIT_EVENT" e "T_PERSISTENT_AUDIT_EVENT_DATA", responsáveis por armazenar informações de registo de *logs* decorrentes do uso da aplicação.

5.4 Pilha tecnológica

Encontrado o método de obtenção de dados de monitorização, foram estudadas as tecnologias que melhor se integram no sentido de produzir uma a prova de conceito da solução proposta. O conjunto de tecnologias terá de se integrar num ambiente já em desenvolvimento, implementado

Implementação da solução

sob a pilha tecnológica JHipster, disponível em [Tea]. Esta pilha oferece uma composição de ferramentas que obedecem aos requisitos não funcionais requeridos (enquanto parte de um projeto europeu, que necessita de articular com outros sistemas/tecnologias). Com esta pilha tecnológica são oferecidas funcionalidades de gestão de utilizadores com diferentes regras de utilização e mecanismos de segurança de autenticação e acesso, *object-relational mapping* (ORM) para acesso a escrita e leitura da base de dados, integração de funcionalidades de criação de rotas para acesso à API e suporte a frameworks de manipulação de *front end* que facilitam o desenvolvimento de um sistema que se pretende dinâmico.

São elementos integrantes da pilha tecnológica o conjunto seguinte de tecnologias/sistemas:

- Java Servlet Apache Tomcat. Software *open-source* que tem o propósito de servir aplicações web desenvolvidas em Java.
- Framework Spring. *Framework* de servidor desenvolvida em Java, que ajuda no desenvolvimento de aplicações de forma simples, rápida e flexível. Possui algumas características como injeção de dependências, padrão orientado a objetos, arquitetura MVC, auxiliada por funcionalidades RESTful fornecidas pelo Hibernate.
- Apache Derby. Sistema de base de dados *open source* relacional, completamente desenvolvido em Java. Apresenta a grande vantagem de apenas precisar de 2.6 *megabytes* para todo o motor de funcionamento e drivers embebidas.
- Framework Angularjs. *Framework* de desenvolvimento web do lado cliente. As suas principais vantagens são a facilidade de *data-binding* entre HTML e Javascript, fácil manipulação de DOM, abordagem orientada ao padrão de desenvolvimento MVC, fácil injeção de dependências e possibilidade de criação de elementos HTML personalizados, chamadas de diretivas. O Angular-translate é um exemplo de uma injeção de dependência, que facilita a criação de aplicações internacionais com várias traduções.
- Grunt, Bower e Karma são, respetivamente, um organizador de tarefas, uma biblioteca de pacotes e uma plataforma de gestão de testes unitários do lado cliente.

A utilização conjunta destas ferramentas deu origem à pilha tecnológica Jhipster, considerada de desenvolvimento ágil, seguro e robusto. A este conjunto de tecnologias juntam-se ainda várias bibliotecas Javascript, consideradas essenciais para a instanciação da solução proposta, na componente de *front end*. Entre estas bibliotecas, destacam-se:

- D3.js, como referenciado na análise do estado da arte, uma das mais poderosas ferramentas para manipulação de DOM.
- NVD3, uma biblioteca que faz uso do D3.js para representar diversos tipos de gráficos.
- Gridster, uma ferramenta de gestão de *dashboards*, fornecendo técnicas de manipulação de dimensões e posições de cada elemento nele contido.

5.5 Desenvolvimento gráfico

A estrutura de desenvolvimento implementa serviços capazes de responder à criação de gráficos independentemente da sua origem, oferecendo ao servidor capacidades de criação de gráficos que guardam informação acerca de opções e dados de forma genérica, podendo ser adicionados a qualquer momento outros tipos de gráficos, podendo ser de monitorização ou não. Tendo em conta a pilha tecnológica estudada no desenho da solução, apresentando o Angular.js como *framework* de desenvolvido do cliente, a solução adota uma implementação o mais genérica possível. Após feita uma análise acerca da existência de bibliotecas capazes de integrar o NVD3 com o Angular conclui-se que não existe nenhuma que realize o tratamento de dados de forma indiferente ao tipo de gráfico. Existem algumas ferramentas capazes de adaptar a biblioteca às funcionalidades do Angular.js, por exemplo o "angular nvd3", no entanto esta apresenta algumas desvantagens como:

- Criação de gráficos com formatação dos dados diferente para cada tipo de gráfico. Esta desvantagem iria traduzir-se em um peso excessivo no tratamento de dados aquando da alteração de um tipo de gráfico;
- Alteração de opções realizada de forma individual, sendo que cada gráfico tem as suas opções formatadas de forma diferente.

Decorrente das necessidades e da oferta que existe atualmente para a tecnologia Angular.js, a solução propõe uma nova biblioteca que estende o uso do NVD3 com características práticas e habituais da *framework* Angular.js.

5.5.1 Angular d3-charts

O desenvolvimento desta biblioteca foi pensado com o propósito de servir, através das funcionalidades presentes na *framework* Angular.js, o propósito da maior parte dos programadores que pretendem uma ferramenta de criação gráfica. Assim, foi desenvolvida uma biblioteca que permite a criação, alteração e remoção de um gráfico através do uso de métodos fornecidos por uma API. Esta nova biblioteca oferece ao programador métodos para interagir com o sistema de forma transparente, executando as mesmas tarefas independente do tipo de gráfico. Esta aplicação fornece capacidades de criação, adição e remoção de funções, alterar opção de personalizações e atualização dos valores dos dados, sendo que estes têm sempre o mesmo formato, colocando o tratamento destes à responsabilidade da biblioteca.

5.6 Desenvolvimento de *dashboards*

O desenvolvimento dos *dashboards* assegura que possam existir vários tipos de serviços, de monitorização ou outro qualquer de visualização. A solução, à semelhança do que acontece com os gráficos, tem a capacidade de identificar o serviço e reagir sob um desenvolvimento comum. A figura 5.4 representa o serviço um esboço da comunicação existente.

Implementação da solução

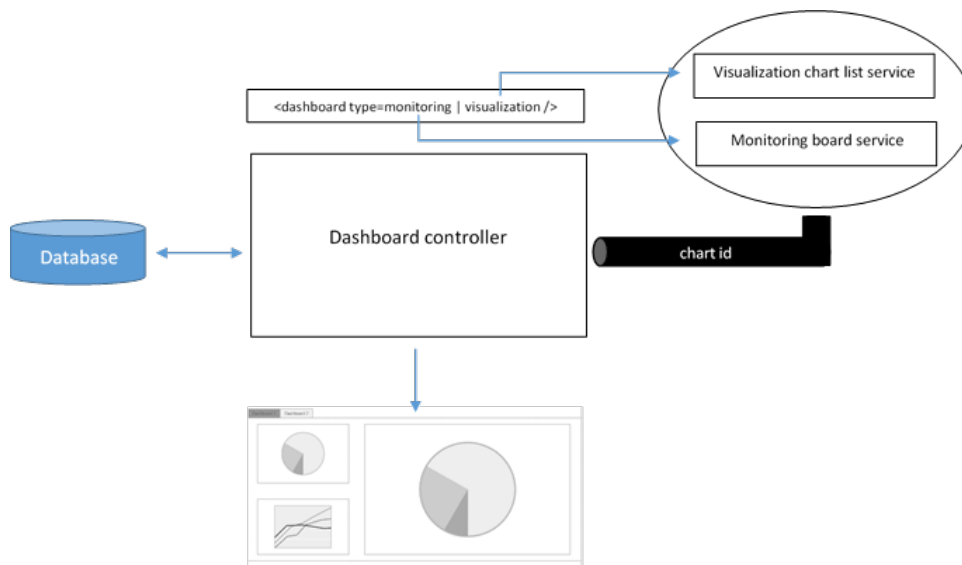


Figura 5.4: Arquitetura de comunicação do módulo dos *dashboards*

Tal como sugere a figura, o Angular.js oferece a possibilidade de criação de elementos HTML, chamadas de diretivas, com diferentes propriedades. No caso ilustrado, dependendo do valor da propriedade "type", o sistema deve subscrever o evento responsável pela adição de gráficos, podendo ser este o serviço pertencer ao ambiente de monitorização ou a outro qualquer serviço que possa ser adicionado ao sistema. A cada adição de um novo gráfico a qualquer desses serviços, o controlador dos *dashboards* deve criar um evento, responsável por receber o ID do gráfico e armazená-lo no seu conteúdo. O controlador dos *dashboards* é posteriormente responsável pelo controlo das suas configurações e posições do gráfico.

5.7 Sumário

O presente capítulo descreve os detalhes de implementação da solução. Para que a solução possa ser integrada num ambiente de produção baseado em Ganglia foi necessário executar teste de avaliação da opção de comunicação mais adequada. Verificou-se que existe a necessidade de uma conjugação de comunicação através de sockets e através da ferramenta Graphite. Também apresenta a arquitetura lógica e o modelo da base de dados, bem como a pilha tecnológica com melhores características para a instanciação da proposta de solução. No fim do capítulo é apresentado o método de desenvolvimento modular, permitindo que a qualquer momento sejam inseridas mais funcionalidades à solução. No capítulo seguinte abordam-se os aspetos de implementação relativos à apresentação da solução estudada.

Capítulo 6

Resultados

O presente capítulo apresenta os resultados dos testes efetuados durante e após a conclusão do desenvolvimento da solução apresentada.

Como referido na contextualização do tema de investigação, a solução de monitorização estudada e proposta irá atuar sobre um sistema *big data* gestor de base de dados. Considera-se essencial um sistema de monitorização, não só para os analistas de rede, mas também para os administradores da infraestrutura e gestores de bases de dados.

É importante reforçar que a implementação da solução proposta da solução deve integrar com outros sistemas previstos no projeto FP7 LeanBigData. Por razões de calendarização do projeto, os testes foram efetuados com um *cluster* de três máquinas, uma vez que a *grid* de 1000 cores não esteve disponível até à altura da escrita deste documento. No entanto, como referido, foram efetuados testes num sistema que possui os mesmos serviços, componentes e métricas que estarão disponíveis no ambiente de produção.

Integrando um projeto europeu de inovação científica, a solução adotada cumpre com os requisitos impostos pelo projeto e pelo sistema a ser monitorado. O desenho da solução foi alvo de processos de melhoria, resultante da avaliação feita por membro do consórcio (administradores de sistemas) para que cumprisse da melhor forma os objetivos pretendidos para o problema. Sempre com uma avaliação integrada ao longo o seu desenvolvimento da solução proposta, o sistema recebeu críticas e sugestões que fizeram com que a solução final fosse a mais completa possível. Por realizar ficam testes de carga com o ambiente de produção. O desenvolvimento da solução foi acompanhada por diversos testes intercalares de prova de conceito, realçando duas demonstrações. Para estas avaliações foi utilizada a estrutura de testes composta por três máquinas.

A primeira apresentação decorreu em Novembro de 2015 com a conclusão parcial das representações da estrutura monitorada. Nesta fase, ao nível da implementação estavam concluídas (mas sujeitas a alterações futuras) a integração com o Ganglia, através do Graphite, a representação de hierarquia, a representação de relação e a janela de componentes selecionados. Nesta

altura, a criação de gráficos servia como prova de conceito, uma vez que ainda não permitia personalização ou associação a *dashboards*. A demonstração da prova de conceito até aí desenvolvida recolheu boas críticas, capacitando a continuação da proposta de solução.

A segunda apresentação decorreu em Março de 2016 e foi feita a um júri de avaliação selecionado pela Comissão Europeia. Já contemplou a representação de dimensão, oferecida pelo gráfico *treemap* e a representação em lista. Nesta altura a implementação da solução proposta já possibilitava a criação de novos *dashboards* e gráficos a eles associados, com as personalizações apresentadas. Decorrente das críticas feitas, foi estudada e concebida uma representação lógica, para uma abordagem diferente daquela que normalmente existe nas soluções atualmente disponíveis de sistemas de monitorização. As críticas obtidas nesta segunda fase continuaram a ser positivas, destacando-se a simplicidade com que o utilizador conseguia navegar na aplicação percorrendo todas as funcionalidades.

Em ambas as apresentações foi possível obter *feedback* de utilizadores do consórcio especializados, que apresentaram opiniões bastante positivas.

A última fase, após Março de 2016 revelou-se importante na realização de testes internos e alterações que possibilitaram afina a implementação da solução. Destacam-se o processo de filtragem dos componentes a apresentar nas vistas de representação da estrutura e pesquisa na vista relacional. Prevê-se que a implementação seja testada com um sistema de 1000 cores, disponível na Universidade Politécnica de Madrid, entre Setembro e Dezembro de 2016.

6.1 Objetivos alcançados

Os problemas levantados foram identificados no capítulo 3.1. Com a proposta de solução apresentada pretendeu-se responder a esses problemas, de modo a obter uma solução capaz de integrar o SGBD desenvolvido no projeto FP7 LeanBigData. Considera-se que a solução proposta responde aos problemas identificados. No entanto, existem sempre aspetos passíveis de serem revistos e melhorados de modo a enriquecer ainda mais a solução.

Para cada um dos problemas identificados a avaliação da solução é a seguinte:

- Oferta de tipos de representação gráfica reduzida, uma vez que quase todos os sistemas limitam-se a representar os dados através gráficos de linhas.
 - A solução proposta oferece um conjunto de soluções gráficas que podem ser usadas na resolução de diferentes objetivos. No entanto, a solução é sempre alvo de melhorias, uma vez que a disponibilização de mais ofertas gráficas nunca é encarada como negativa.
- Pouca personalização gráfica, no que diz respeito a cores, títulos, descrições e legendas.
 - Na presente proposta foram adotadas todas as personalizações idealizadas para os elementos, respondendo à problemática de forma eficiente.

Resultados

- Atualização de gráficos corresponde a transferir imagens, exigindo novas imagens para cada atualização.
 - Com o uso do Graphite e das ferramentas de criação gráfica do lado cliente, a solução cumpre com o objetivo.
- Representação limitada da estrutura do sistema monitorizado, baseando-se em listas de navegação em profundidade, o que dificulta a localização de componentes.
 - Foram apresentadas cinco vistas alternativas da representação da estrutura da rede, disponibilizando ao utilizador diferentes perspetivas para diferentes casos de análise.
- Representação orientadas aos componentes físicos do sistema monitorado e não à interpretação lógica dos serviços existentes nesse sistema.
 - Realização da vista lógica que permite a geração de *dashboards* orientados aos serviços.
- *Dashboards* com manipulação reduzida.
 - Com o uso da ferramenta Gridster, a solução consegue responder à problemática encontrada. No entanto sujeita a melhorias, uma vez que uma solução mais abrangente poderia oferecer a liberdade total de manipulação de dimensão, posição e sobreposição caso fosse pretendido.
- Organização complexa e por vezes de difícil interpretação.
 - A formulação de uma aplicação *single page* oferece uma simplicidade extra, facilitando a navegação e exploração do sistema sem necessidade de uma aprendizagem complexa.

6.2 Artigos científicos

Da investigação realizada, resultaram dois artigos científicos. O primeiro com o objetivo de avaliação intercalar do trabalho realizado e o segundo já com a totalidade do estudo.

O primeiro foi escrito durante a primeira fase de desenvolvimento e foi aceite e apresentado na Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI) a 16 de Junho de 2016. Encontra-se publicado nas atas da conferência, na área da interação pessoa-computador (anexo B deste documento). Decorrente da apresentação, surgiram críticas construtivas para trabalho futuro e uma avaliação geral muito positiva. O segundo artigo, escrito e submetido já na fase final de avaliação, foi realizado com o objetivo de apresentação na conferência INForum 2016, encontrando-se à data submetido e em fase de avaliação.

Considera-se que o estudo realizado conclui-se com bons resultados, espectáveis, respondendo ao problemas e conseguindo inovar sobre os sistemas de monitorização habitualmente usados.

Resultados

Capítulo 7

Conclusões

São várias as empresas que usam as informações dos seus utilizadores para se tornarem mais competitivas. Para o processamento desta informação, são usados sistemas distribuídos agindo de forma paralela. Em sistemas com esta configuração, a informação é armazenada e processada em *clusters* com centenas de máquinas, onde naturalmente, um sistema de monitorização verifica-se obrigatória, contribuindo para a análise do desempenho de cada componente do sistema, prevenindo eventuais problemas e até mesmo falhas.

A revisão de algumas das ferramentas de monitorização utilizadas nos dias de hoje permite concluir que não existe uma ferramenta que integre as melhores soluções de recolha, processamento, armazenamento e visualização dos dados monitorizados. Da comparação das ferramentas, conclui-se que o Ganglia apresenta mecanismos de desempenho e escalabilidade extraordinários, usufruindo de recolha de dados em redes complexas e armazenamento em ficheiros RRD. No entanto, a oferta de visualização dos dados apresenta problemas. São estes: representação de dados das métricas realizado unicamente através de gráficos de linhas; personalização gráfica limitada a mudança de cores de funções nos gráficos; uso da ferramenta Graphite para geração gráfica, criando sobrecargas na rede decorrentes do envio sucessivo de atualizações em formato de imagem do servidor para o cliente; pesquisa e localização de elementos através do nome ou listas respetivamente, revelando-se um processo demorado com o desconhecimento e crescimento do sistema monitorado. Conclui-se que complementando o Ganglia com uma nova interface, é possível apresentar uma solução completa, escalável e útil no reconhecimento de problemas e causas de erro.

O presente tema de estudo propõe soluções visuais capazes de responder a estes problemas de forma mais eficiente, nomeadamente: oferta de diferentes tipos de gráficos com capacidades de responder a necessidades diferentes; opções de personalização gráfica através da identificação dos gráficos por títulos e legendas, com possibilidade de alteração das cores das funções ou nomes dos eixos; construção gráfica efetuada do lado do cliente, recorrendo a ferramentas poderosas existentes hoje em dia, diminuindo assim a largura de banda usada para atualizações de valores; Oferta

de vistas representativas da estrutura do sistema, auxiliadas de técnicas de manipulação gráfica, promovendo a rápida identificação de componentes do sistema; Disponibilização de *dashboards* compostos por gráficos apropriados a cada elemento, onde facilmente se pode retirar e adicionar novos gráficos e organizá-los de forma dinâmica em tamanho e posição.

A solução foi submetida a avaliações realizadas pelo projeto FP7 LeanBigData e pela Comissão Europeia, resultando em observações de objetivos cumpridos e aprovação do sistema proposto. O presente trabalho foi também sujeito a uma avaliação pela comunidade científica, realizado através da realização de dois artigos científicos para conferências da área (encontrando-se à data, um apresentado e outro em fase de avaliação).

A solução proposta beneficia das vantagens presentes em alguns componentes de monitorização do Ganglia, corrigindo e melhorando os défices existentes na componente de visualização, possibilitando ao utilizador um maior conhecimento do sistema e estrutura monitorizada, tornando o processo de deteção de causas de erro mais eficiente.

7.1 Trabalho Futuro

A proposta de solução para o problema da representação gráfica de dados de monitorização revelou-se cumprida. No entanto, existem sempre progressos que têm capacidade de oferecer resultados ainda melhores. Com base no estado atual das ferramentas de monitorização presentes no mercado, existem representações georreferenciadas que podem acrescentar valor à análise do sistema. Sistemas de alertas e eventos são também um foco importante que complementaria a interface visual mais objetiva. O uso do Ganglia como ferramenta de interação com o sistema operativo e armazenamento dos respetivos dados recolhidos conclui-se eficaz. No entanto, o auxílio da sua configuração através da interface, ao invés dos ficheiros atualmente existentes, resultaria em uma configuração simplificada do sistema. A extensão das possibilidades gráficas de representação de dados será sempre um aspeto diferenciador e passível de ser explorado. Por fim, existem manipulações da componente dos *dashboards* que pode melhorar a sua utilização, tal como o uso de profundidade na organização dos seus componentes.

Referências

- [ADBF⁺05] Sergio Andreato, Natascia De Bortoli, Sergio Fantinel, Antonia Ghiselli, Gian Luca Rubini, Gennaro Tortone e Maria Cristina Vistoli. Gridice: a monitoring service for grid systems. *Future Generation Computer Systems*, 21(4):559–571, 2005.
- [BHP02] Franck Bonnassieux, Robert Harakaly e Pascale Primet. Mapcenter: an open grid status visualization tool. In *Proceedings of the ICSC 15th International Conference on Parallel and Distributed Computing Systems*, 2002.
- [Bla13] Chris Blake. Observium, the do-it-all monitoring application, April 2013. Acedido pela última vez em 27 de Julho de 2016. URL: <http://servernetworktech.com/2013/04/observium-the-do-it-all-monitoring-application/>.
- [BMMS91] Andreas Buja, John Alan McDonald, John Michalak e Werner Stuetzle. Interactive data visualization using focusing and linking. In *Visualization, 1991. Visualization '91, Proceedings., IEEE Conference on*, pages 156–163. IEEE, 1991.
- [Bos11] Mike Bostock. D3, data-driven documents, 2011. Acedido pela última vez em 27 de Julho de 2016. URL: <http://d3js.org/>.
- [CM] Inc Chad Maynard, TechOptics. Acedido pela última vez em 27 de Julho de 2016. URL: <http://www.observium.org/>.
- [Öd15] Torkel Ödegaard. Grafana, 2015. Acedido pela última vez em 27 de Julho de 2016. URL: <http://grafana.org/>.
- [Dav] Chris Davis. The architecture of open source applications. Acedido pela última vez em 27 de Julho de 2016. URL: <http://aosabook.org/en/graphite.html>.
- [DLP09] Aba-Sah Dadzie, Vitaveska Lanfranchi e Daniela Petrelli. Seeing is believing: Linking data with knowledge. *Information Visualization*, 8(3):197–211, 2009.
- [Doc] Observium Docs. Observium. Acedido pela última vez em 27 de Julho de 2016. URL: <http://www.observium.org/docs/>.
- [Duc] Ducksboard. Gridster.js. Acedido pela última vez em 27 de Julho de 2016. URL: <http://gridster.net/>.
- [Ell14] Justin Ellingwood. An introduction to tracking statistics with graphite, statsd, and collectd, May 2014. Acedido pela última vez em 27 de Julho de 2016. URL: <https://www.digitalocean.com/community/tutorials/an-introduction-to-tracking-statistics-with-graphite-statsd-and-collectd>.

REFERÊNCIAS

- [Gan01] Ganglia. Ganglia monitoring systems, January 2001. Acedido pela última vez em 27 de Julho de 2016. URL: <http://ganglia.sourceforge.net/>.
- [Gro] The Cacti Group. About cacti. Acedido pela última vez em 27 de Julho de 2016. URL: <http://www.cacti.net/>.
- [GWB⁺04] Michael Gerndt, Roland Wismüller, Zoltán Balaton, Gábor Gombás, Péter Kacsuk, Zsolt Németh, N Podhorecki, Hong-Linh Truong, Thomas Fahringer, Marian Bubak et al. *Performance Tools for the Grid: State of the Art and Future: Apart White Paper*. Shaker, 2004.
- [Hea96] Christopher G Healey. Choosing effective colours for data visualization. In *Visualization'96. Proceedings.*, pages 263–270. IEEE, 1996.
- [HMM00] Ivan Herman, Guy Melançon e M Scott Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.
- [IS11] Noah Iliinsky e Julie Steele. *Designing Data Visualizations: Representing Informational Relationships*. "O'Reilly Media, Inc.", 2011.
- [Jam14] Josh James. Data never sleeps 2.0, Abril 2014. Acedido pela última vez em 27 de Junho de 2016. URL: <https://www.domo.com/blog/2014/04/data-never-sleeps-2-0/>.
- [Jos15] Charles Joseph. Open source server monitoring tools – cacti, zabbix, nagios, August 2015. Acedido pela última vez em 27 de Julho de 2016. URL: <https://cloudstats.me/2015/08/08/open-source-server-monitoring-tools-cacti-zabbix-nagios/>.
- [Kun] Gion Kunz. Chartlist.js. Acedido pela última vez em 27 de Julho de 2016. URL: <http://gionkunz.github.io/chartist-js/>.
- [LDR03] DongWoo Lee, Jack J Dongarra e Rudrapatna S Ramakrishna. Visperf: Monitoring tool for grid computing. In *Computational Science—ICCS 2003*, pages 233–243. Springer, 2003.
- [Lea14] LeanBigData. Lean big data - do it faster with less resources, 2014. Acedido pela última vez em 27 de Julho de 2016. URL: <http://leanbigdata.eu/>.
- [LNV⁺09] Iosif Legrand, Harvey Newman, Ramiro Voicu, Catalin Cirstoiu, Costin Grigoras, Ciprian Dobre, Adrian Muraru, Alexandru Costan, Mihaela Dediu e Corina Stratan. Monalisa: An agent based, dynamic service system to monitor, control and optimize distributed systems. *Computer Physics Communications*, 180(12):2472–2498, 2009.
- [Mar07] Nico Marrero. Visualization metrics: An overview. 2007.
- [MCC04] Matthew L Massie, Brent N Chun e David E Culler. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(7):817–840, 2004.
- [MCR90] Jock D Mackinlay, Stuart K Card e George G Robertson. Rapid controlled movement through a virtual 3d workspace. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 171–176. ACM, 1990.

REFERÊNCIAS

- [Mona] MonALISA. Monitoring agents using a large integrated services architecture. Acedido pela última vez em 27 de Julho de 2016. URL: <http://monalisa.caltech.edu/monalisa.htm>.
- [Monb] INC MongoDB. Monitoring for mongodb. Acedido pela última vez em 27 de Julho de 2016. URL: <https://docs.mongodb.org/manual/administration/monitoring/>.
- [NLG⁺03] Harvey B Newman, Iosif C Legrand, Philippe Galvez, Ramiro Voicu e Catalin Cirstoiu. Monalisa: A distributed monitoring service architecture. *arXiv preprint cs/0306096*, 2003.
- [NSC05] Marcelo Veiga Neves, Tiago Scheid e Andrea Schwertner Charao. Monitoraç ao de clusters com a ferramenta ganglia: avaliaç ao e adaptaç ao. 2005.
- [Par] Novus Partners. Re-usable charts for d3.js. Acedido pela última vez em 27 de Julho de 2016. URL: <http://nvd3.org/>.
- [PMDC09] Daniela Petrelli, Suvodeep Mazumdar, Aba-Sah Dadzie e Fabio Ciravegna. *Multi visualization and dynamic query for effective exploration of semantic data*. Springer, 2009.
- [RW] Custodian DataCentres Robert Williams, Technical Director. Acedido pela última vez em 27 de Julho de 2016. URL: <http://www.observium.org/>.
- [SC00] Sônia Fernandes Silva e Tiziana Catarci. Visualization of linear time-oriented data: a survey. In *Web Information Systems Engineering, 2000. Proceedings of the First International Conference on*, volume 1, pages 310–319. IEEE, 2000.
- [SC15] SR Sreeja e Sangita Chaudhari. Study on grid resource monitoring and prediction. *Procedia Computer Science*, 45:815–822, 2015.
- [SH09] Ed Simmonds e Jason Harrington. Scf/fef evaluation of nagios and zabbix monitoring systems. July 2009.
- [Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.
- [Sol] Qlik Solutions. Qlikview. Acedido pela última vez em 27 de Julho de 2016. URL: <http://www.qlik.com/products/qlikview>.
- [Sol09] Nagios Solutions. Nagios, 2009. Acedido pela última vez em 27 de Julho de 2016. URL: <https://www.nagios.org/about/>.
- [Swi12] Austin Graves Swinney. Ganglia vs collectd vs. cacti? advantage and disadvantage, November 2012. Acedido pela última vez em 27 de Julho de 2016. URL: <https://www.quora.com/Ganglia-vs-collectd-vs-cacti-advantage-and-disadvantage>.
- [Sys] Pandora FMS: Flexible Monitoring System. Pandora fms. Acedido pela última vez em 27 de Julho de 2016. URL: <https://sourceforge.net/projects/pandora/>.

REFERÊNCIAS

- [Tea] Jhipster Team. Java hipster. Acedido pela última vez em 27 de Julho de 2016. URL: <https://jhipster.github.io/>.
- [TGM83] Edward R Tufte e PR Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.
- [TSA⁺10] Ashish Thusoo, Zheng Shao, Suresh Anthony, Dhruba Borthakur, Namit Jain, Joydeep Sen Sarma, Raghotham Murthy e Hao Liu. Data warehousing and analytics infrastructure at facebook. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1013–1020. ACM, 2010.
- [VDFL⁺00] Andries Van Dam, Andrew S Forsberg, David H Laidlaw, Joseph J LaViola Jr e Rosemary M Simpson. Immersive vr for scientific visualization: A progress report. *Computer Graphics and Applications, IEEE*, 20(6):26–52, 2000.
- [Wik15a] Wikipedia. Abacus, December 2015. Acedido pela última vez em 27 de Julho de 2016. URL: <https://en.wikipedia.org/wiki/Abacus>.
- [Wik15b] Wikipedia. Comparison of network monitoring systems, December 2015. Acedido pela última vez em 27 de Julho de 2016. URL: https://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems.
- [Wik15c] Wikipedia. Graphs and charts, November 2015. Acedido pela última vez em 27 de Julho de 2016. URL: https://en.wikipedia.org/wiki/Wikipedia:Graphs_and_charts.
- [Won] Edmund Wong. Network monitoring fundamentals and standards. Acedido pela última vez em 27 de Julho de 2016. URL: http://www.cse.wustl.edu/~jain/cis788-97/ftp/net_monitoring/.
- [WT06] Al Weber e Ron Thomas. Key performance indicators: Measuring and managing the maintenance function. *Ivara Corporation*, 2006.
- [Xu11] Tian Xu. Grid monitoring system survey. 2011.
- [XWG⁺13] Jing Xia, Feiran Wu, Fangzhou Guo, Cong Xie, Zhen Liu e Wei Chen. An online visualization system for streaming log data of computing clusters. *Tsinghua Science and Technology*, 2:011, 2013.
- [Zuc15] Mark Zuckerberg. Q&A facebook, Junho 2015. Acedido pela última vez em 27 de Julho de 2016. URL: <https://www.facebook.com/zuck/posts/10102213601037571>.

Anexo A

Casos de uso

Tabela A.1: Caso de uso 1.1

| | |
|---------------|--|
| Identificador | 1.1 |
| Tipo | Configuração |
| Especificação | Como utilizador pretendo introduzir o nome de utilizador e palavra-chave para ter acesso à aplicação |
| Prioridade | 5 |
| Pré-condição | Estar registado na aplicação |

Tabela A.2: Caso de uso 1.2

| | |
|---------------|--|
| Identificador | 1.2 |
| Tipo | Configuração |
| Especificação | Como utilizador pretendo modificar a minha palavra chave |
| Prioridade | 4 |
| Pré-condição | A.1 |

Tabela A.3: Caso de uso 1.3

| | |
|---------------|---|
| Identificador | 1.3 |
| Tipo | Configuração |
| Especificação | Como utilizador pretendo especificar os endereços dos servidores de Ganglia, Graphite, Zookeeper e o caminho do ficheiro onde está descrita a organização dos componentes do sistema. |
| Prioridade | 4 |
| Pré-condição | A.1 |

Casos de uso

Tabela A.4: Caso de uso 2.1

| | |
|---------------|---|
| Identificador | 2.1 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo visualizar a estrutura do sistema para ter conhecimento da composição deste. |
| Prioridade | 5 |
| Pré-condição | A.1 e A.3 |

Tabela A.5: Caso de uso 2.2

| | |
|---------------|--|
| Identificador | 2.2 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo pesquisar um componente pelo nome para assegurar uma pesquisa rápida e eficiente. |
| Prioridade | 4 |
| Pré-condição | A.1 e A.3 |

Tabela A.6: Caso de uso 2.3

| | |
|---------------|---|
| Identificador | 2.3 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo criar novos gráficos, compostos por métricas escolhidas previamente, para obter informações dos valores medidos. |
| Prioridade | 5 |
| Pré-condição | A.1 e A.3 |

Tabela A.7: Caso de uso 2.4

| | |
|---------------|--|
| Identificador | 2.4 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo personalizar os gráficos para obter uma melhor interpretação e diferenciação entre os vários gráficos apresentados. |
| Prioridade | 4 |
| Pré-condição | A.1 e A.6 |

Casos de uso

Tabela A.8: Caso de uso 2.5

| | |
|---------------|--|
| Identificador | 2.5 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo criar <i>dashboards</i> para organizar os gráficos. |
| Prioridade | 5 |
| Pré-condição | A.1 |

Tabela A.9: Caso de uso 2.6

| | |
|---------------|--|
| Identificador | 2.6 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo personalizar <i>dashboards</i> para beneficiar de uma melhor interpretação. |
| Prioridade | 4 |
| Pré-condição | A.1 e A.8 |

Tabela A.10: Caso de uso 2.7

| | |
|---------------|---|
| Identificador | 2.7 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo eliminar gráficos para que estes não voltem a surgir no <i>dashboard</i> . |
| Prioridade | 5 |
| Pré-condição | A.1 e A.6 |

Tabela A.11: Caso de uso 2.8

| | |
|---------------|---|
| Identificador | 2.8 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo eliminar <i>dashboards</i> para que estes não voltem a surgir. |
| Prioridade | 5 |
| Pré-condição | A.1 e A.8 |

Casos de uso

Tabela A.12: Caso de uso 2.9

| | |
|---------------|---|
| Identificador | 2.9 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo modificar em tamanho e posição os gráficos em um <i>dashboard</i> para obter uma melhor organização. |
| Prioridade | 4 |
| Pré-condição | A.1 , A.6 e A.8 |

Tabela A.13: Caso de uso 2.10

| | |
|---------------|---|
| Identificador | 2.10 |
| Tipo | Funcionalidades |
| Especificação | Como utilizador pretendo criar <i>dashboards templates</i> com base na informação presente nos ficheiros de organização dos componentes do sistema. |
| Prioridade | 3 |
| Pré-condição | A.3 |

Anexo B

Artigo CISTI'2016

Sistemas de Monitorização para Sistemas Paralelos e Distribuídos de Controlo de Dados

Monitoring Systems for Parallel Distributed Data Management Systems

Emanuel Pinho^{1,2}

emanuel.a.pinho@inescporto.pt

¹ DEI/FEUP Rua Dr. Roberto Frias,
4200-465 Porto, Portugal

Alexandre Valle de Carvalho^{1,2}

alexandre.carvalho@inesctec.pt

² INESC TEC Campus da FEUP, R. Dr. Roberto Frias,
4200-465 Porto, Portugal

Resumo — Vulgarmente, um sistema de informação BigData está acompanhado de um sistema de monitorização para avaliação de performance e prevenção de erros. Na atualidade as interfaces destes sistemas apresentam desvantagens como a limitação gráfica e a sua abordagem direcionada a componentes físicos. O principal objetivo é estudar mecanismos visuais gráficos e de interação que permitam a representação de dados de monitorização de performance em ambientes de *grid computing*, permitindo fornecer ao utilizador final informações capazes de contribuir de forma objetiva na perceção da análise comportamental do sistema. A participação na conferência tem como objetivo apresentar o estado da arte do tópico de estudo e realizar uma avaliação intercalar do trabalho realizado bem como da proposta de trabalho futuro.

Palavras Chave – *grid computing; monitorização; interação pessoa-computador; gráficos; visuais.*

Abstract — Usually, a Big Data system has a monitoring system for performance evaluation and error prevention. Although, there are some disadvantages in the way that these tools display the information and its targeted approach to physical components. The main goal is to study visual and interaction mechanisms that allow the representation of monitoring data in grid computing environments, providing the end-user information which can contribute objectively to the system analysis. This paper has the purpose to present the state of the art, carries out an intermediate evaluation of the current work and present the proposed solution.

Keywords – *grid computing; monitoring; human-machine interaction; graphical; visual.*

I. AS PESSOAS E OS DADOS QUE PRODUZEM

Hoje em dia, a quantidade de informação gerada eletronicamente faz com que sistemas de bases de dados estejam em constante crescimento [1]. A dimensão da informação que produzimos é realmente enorme e cresce de dia para dia. De acordo com McAfee et al. [2], no ano de 2012, cerca de 2.5 *exabytes* de dados foram criados diariamente. Estes dados são utilizados por grandes empresas mundiais na área das tecnologias da informação como principal estratégia de negócio, fazendo com que a sua análise e tratamento, sejam fatores decisivos na perceção das necessidades dos seus clientes. Mark Zuckerberg refere-o em [3]. Estes dados são processados em sistemas complexos, distribuídos e paralelos para que o seu desempenho seja o melhor [4]. No entanto, os efeitos consequentes do seu uso pode ser um inconveniente. De acordo com Andreozzi et al [5], o uso de sistemas distribuídos obriga à

existência de coordenação de recursos e serviços entre estes, trazendo a consequente variação de comportamentos e decisões. Estas variações têm efeitos repercutivos no desempenho de cada componente do sistema, transformando a monitorização destes componentes numa tarefa obrigatória na sua manutenção. É com esta necessidade que surgem os sistemas de monitorização.

II. SISTEMAS DE MONITORIZAÇÃO

Os Sistemas de monitorização são imprescindíveis para qualquer organização que possua componentes que realizam tarefas críticas. Empresas como a MongoDB já o afirmaram em [6], caracterizando a monitorização como um componente crítico em toda a administração de sistemas de bases de dados. A sua principal tarefa traduz-se na análise do estado do sistema, subcategorizando-se na sua análise comportamental, deteção de anomalias, prevenção de falhas e melhoria de desempenho. Xu [7] refere que a manipulação de sistemas potencialmente heterogéneos requerem ferramentas capazes de capturar informação, disponibilizada em tempo real. Sreeja et al. [8] afirma que a deteção de falhas através das ferramentas de monitorização é um processo obrigatório para poder recorrer a mecanismos de recuperação.

O método de atuação dos sistemas de monitorização pode ser dividido em recolha de dados, processamento e armazenamento e finalmente visualização. A primeira tarefa de qualquer sistema que pretenda disponibilizar dados formatados é garantir o acesso a esses dados. A recolha de dados a partir de sensores é, portanto, a primeira tarefa a executar, podendo ser centralizada, distribuída ou até apenas local. O segundo estado passa pelo processamento dos dados e respetivo armazenamento. A terceira fase corresponde à interação com o sistema de armazenamento de forma a representar graficamente a informação. Coerente com Silva e Catarci, em [9], os dados recolhidos através dos sensores são representados sob forma linear e temporal. No entanto, existem outros métodos gráficos passíveis de transcreverem os dados num sistema distribuído. A estrutura do sistema é um exemplo onde a formatação pode adquirir formas hierárquicas ou representação georreferenciada.

III. DEFINIÇÃO DO PROBLEMA

Este trabalho de investigação incide precisamente na última etapa, a visualização. Estas ferramentas apresentam atualmente desempenhos notáveis tanto na recolha como no processamento

e armazenamento da informação. No entanto, existem aspetos visuais débeis que devem ser revistos.

É importante lembrar que a principal função destes sistemas é providenciar visualmente resultados onde é rápido identificar problemas que podem causar falhas. A diferenciação é, portanto, um fator indispensável na sua aparência. O sistema tem de ser capaz de realçar os aspetos que merecem a atenção do utilizador.

A escalabilidade é outros dos problemas que ocorre num sistema complexo e pode ser um *handicap* à representação esquemática deste sistema. A sua representação visual deve permanecer clara, objetiva, não ambígua e intuitiva. Coerente com esta dificuldade, a usabilidade deve também ser garantida, não permitindo que as variações de dimensão do sistema interfiram com a forma de pesquisa e de deteção de componentes alvos.

IV. FERRAMENTAS DE MONITORIZAÇÃO

Tendo em conta as ferramentas estudadas é possível constatar que o mercado está repleto de ferramentas de monitorização e que existem debilidades, tornando muitas vezes a deteção da origem do problema uma tarefa difícil. Nesta secção são aprofundadas algumas dessas ferramentas, bem como identificadas as respetivas vantagens e deficiências.

A. Ganglia

Ganglia é definido por [10] como uma ferramenta escalável e distribuída para sistemas com elevado desempenho, como são os casos de *clusters* e *grids*. Segundo Andreozzi et al. [5], um sistema completo *ganglia* contém três componentes: *Gmond*, *Gmetad*, *Gweb*. *Gmond* é responsável pela aquisição de valores dos sensores, interagindo com o sistema operativo. *Gmetad* é o componente que faz a ligação entre *Gmond* e os ficheiros de armazenamento *RRD*. De acordo com Massie et al. [11], os ficheiros *RRD* fornecem uma alocação dinâmica em vários segmentos temporais, mantendo constante a dimensão da base de dados. Este é um dos mais usados métodos de armazenamento de dados temporais. O terceiro componente, e com maior relevância para o presente estudo, é o *Gweb*. Fornece dados visuais representativos do conteúdo presente nos ficheiros *RRD*. Segundo Massie et al. em [12], o *Ganglia Web Client* comunica diretamente com o *Graphite*, que é responsável por adquirir a informação dos ficheiros *RRD* e transformá-los em gráficos *PNG*, enviando-os para o cliente. Os autores também referem que a relação entre estes componentes, desenhados de forma hierárquica, representa a maior vantagem face a outras ferramentas.

B. Observium

Observium é definido por [13] como uma ferramenta de monitorização com deteção automática do ambiente. Desta forma, esta é capaz de se adaptar ao sistema, através do uso de *SNMP*. Chris Blake refere que isso pode ser uma vantagem para utilizadores que não pretendem perder muito tempo com configurações, mas por outro lado tem mais limitações, como é o exemplo da extensão de métricas a analisar [14]. Tal como o *Ganglia*, esta ferramenta faz uso do formato de ficheiros *RRD* para armazenar os valores recolhidos pelos sensores.

C. Nagios

Nagios é mais uma ferramenta *open-source* que contém, segundo [15], características normais de qualquer sistema de

monitorização, como serviços de rede, captura de recursos por diferentes máquinas, sistema de alertas e representação gráfica através de *plugins* de *charting*. No entanto, como afirmado por Andreozzi et al. [5], o maior problema é ter um raio de ação pequeno, atuando apenas em redes locais.

D. Zabbix

É capaz de monitorizar um sistema distribuída de complexidade elevada, à semelhança do *ganglia*, mas com mecanismos de construção automática da rede. As características de maior relevância são: a recolha de dados de protocolos como *http*, *ftp*, *ssh*, *pop3*, *smtp*, etc.; mecanismos de autenticação e permissões baseadas em *rules*; notificações personalizadas; *Audit logs*. No entanto, e tal como refere Joseph em [16], quanto mais complexo em termos de organização visual, mais tempo demora o utilizador a adaptar-se à aplicação. O *Zabbix* também apresenta mecanismos de configuração pobres, sendo algumas das configurações apenas editáveis em ficheiros *xml* [17].

V. TÉCNICAS DE VISUALIZAÇÃO

Considerando o foco principal deste projeto, isto é, a visualização dos dados monitorizados, existem técnicas de visualização que podem ajudar a uma mais fácil perceção de problema e ajustar o foco de atenção do utilizador. As técnicas de representação gráfica de diagramas são identificadas como ilustrações de conjuntos de dados [18]. Estas técnicas, segundo Iliinsky e Steele em [19], são excelentes para captar atenção cerebral e desta forma conseguir ser mais objetivas na transmissão de informação. Dam et al. [20] afirma que mais de 50% dos neurónios humanos são dedicados à visão, destacando a excecional capacidade de reconhecimento de padrões.

Comparando um sistema de monitorização a uma história, [21] refere que o principal objetivo de ambos é transmitir algo a alguém, portanto, é necessário que o emissor se coloque na pele do recetor, usando a “linguagem” apropriada na comunicação. Depois, o emissor tem de perceber qual a estrutura da informação que pretende transmitir, nomeadamente se são temporais ou organizações hierárquicas, quantas variáveis são identificáveis, qual o grau de variação, etc.

À partida é possível identificar algumas características úteis na representação visual de dados de monitorização. Healey [22] identifica a cor como uma variável visual importante e frequentemente usada para diferenciação de pormenores. Buja et al [23] e Mackinlay et al. [24] salvaguardam técnicas de *zooming* e manipulação de *layout* como fator de seleção de pontos de interesse.

VI. PROPOSTA DE SOLUÇÃO

Depois de estudadas as ferramentas e com o foco definido na visualização e interface com o utilizador, verificou-se que o *Ganglia* apresenta características de captura e armazenamento excelentes quando aplicados a sistemas complexos, distribuídos e facilmente escaláveis, sendo adotada esta ferramenta para executar essas tarefas. É proposto um sistema com os componentes *Gmond* e *Gmetad*, que por sua vez é auxiliado por uma instância do *Graphite* como forma de estabelecer a ponte entre os dados de armazenamento *RRD* produzidos e a interface *web client*.

Analisando as ferramentas abordadas, é possível verificar que existem aspetos em comum passíveis de serem melhorados: a representação gráfica de dados lineares e temporais, apesar de ser representada logicamente em gráficos de linhas, é passível de ser sujeita a outras técnicas de visualização. Existem alternativas que beneficiam a capacidade de extração de conhecimento pelo utilizador. A representação das métricas pode ser auxiliada por gráficos *parallel coordinates* para relação de componentes e comparação de comportamentos, gráficos de tempo real de *gauge* ou até mesmo gráficos circulares para representação percentual da ação de cada componente no sistema. A customização gráfica é um aspeto que contribui para a capacidade de diferenciação. Não só as cores das funções são importantes, mas aspetos como título, legendas, *labels* de identificação ou mudança de nomes dos eixos podem ser usados pelos utilizadores para personalização de aspetos diferenciadores; O uso do *Graphite* como método de comunicação com dados *RRD* fornece serviços de comunicação RESTful interessantes do ponto de vista de comunicação. No entanto, o envio de novos gráficos em formato de imagem a cada atualização de valores cria um *overhead* desnecessário na rede. A proposta de solução recomenda que a construção gráfica seja efetuada do lado do cliente para que apenas dados de atualização sejam transferidos; Os métodos de pesquisa e localização de componentes podem-se tornar tarefas difíceis quando o sistema se depara com problemas de escalabilidade. Como forma de resolver estas situações, a proposta de solução fornece gráficos de representação estrutural que facilitam estas pesquisas, através de representações em *tree charts*, *hiveplots* [25] para demonstração das conexões entre componentes ou treemaps para fornecer noção de profundidade e dimensão do sistema. A organização e construção de *dashboards* é outra interação que pode ser melhorada. É relevante que o sistema providencie métodos de ajuste dimensional e ordenação de gráficos de forma *user-friendly*. Uma ferramenta capaz de criar *dashboards* por defeito, direcionados a componentes em específico pode ajudar na facilidade de identificação de problemas; Podemos também observar que atualmente as ferramentas se focam na representação de métricas física, como é o exemplo das análises de *cpu*, *memória*, etc. A nova proposta foca-se na sua visualização lógica, onde podemos obter e relacionar camadas de diferentes serviços (query engines, persistency layers, storage layers, etc) que existem nas máquinas e assim analisar melhores alternativas de distribuição.

VII. TRABALHO REALIZADO

A aplicação *web* assenta numa estrutura composta por um *cluster* de três máquinas sujeitas a monitorização, sendo centralizada a informação correspondente a todas as métricas lidas num único servidor que aloja o *Gmetad*. Este servidor aloja igualmente uma instância do *graphite* que comunica com o cliente. A estrutura pode ser observada na figura 1.

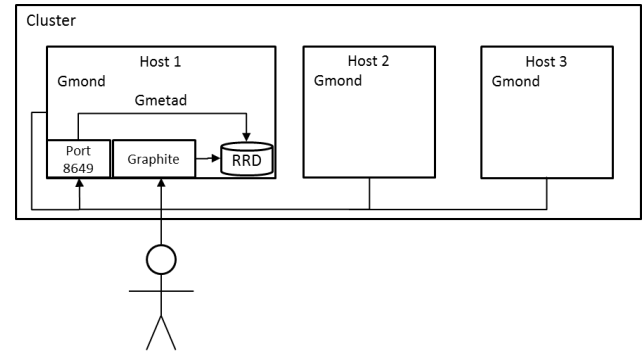


Figure 1. Estrutura do sistema a monitorizar

A construção gráfica é efetuada do lado cliente, utilizando a tecnologia *D3.js* [26], que não sendo diretamente uma ferramenta de criação gráfica, oferece métodos de manipulação de objetos DOM, que possibilitam a construção de qualquer género de diagramas de representação de dados.

A representação geral da estrutura do sistema é um ponto essencial para a compreensão e fácil identificação de componentes. O sistema contém uma estrutura hierárquica, que segue um padrão ótimo de desenho em árvore, tal como representado na figura 2.

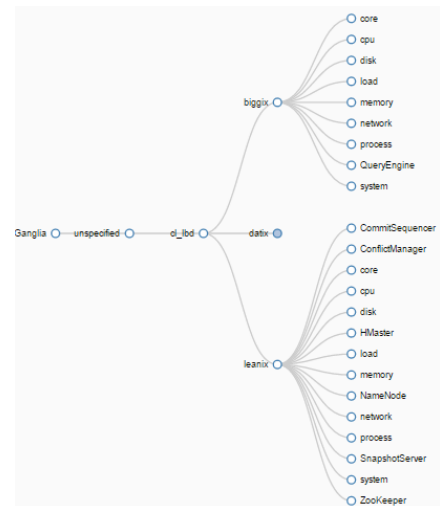


Figure 2. Representação do sistema em gráfico hierárquico

Para rápida identificação da origem do problema ou como forma de obter possíveis soluções, é útil oferecer uma vista lógica, capaz de mostrar a relação entre os componentes. Para isso foi desenvolvido um gráfico *hiveplot*, que possui três eixos que representam as métricas, famílias e máquinas. É possível observar na figura 3 que existem ligações que representam as relações entre cada componente. Por exemplo, no caso apresentado, em que se encontra selecionada a família *cpu*, são realçadas as ligações às métricas que esta contém e quais os *hosts* que a analisam.

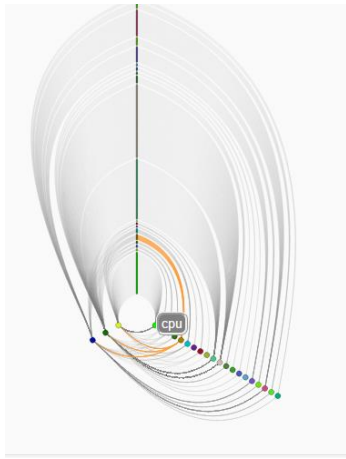


Figure 3. Representação do sistema com hiveplot

O sistema atual contém ainda uma representação sob forma de *treemap*, que contribui para a noção de dimensão do sistema e de cada um dos seus componentes, podendo ser observada na figura 4, com cada área representando uma máquina do sistema.

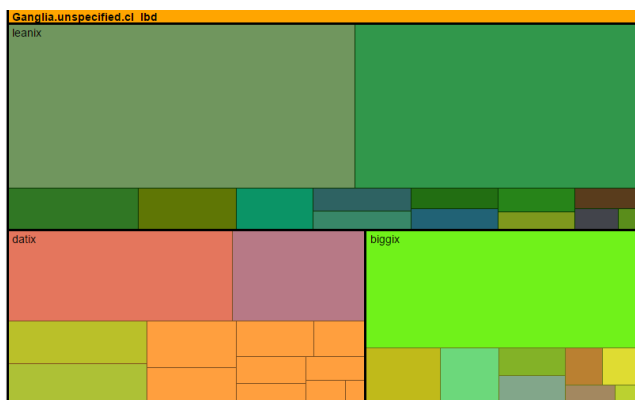


Figure 4. Representação do sistema com *treemap*

VIII. CONCLUSÕES E TRABALHO FUTURO

A aplicação em desenvolvimento contém aspetos visuais facilitadores do trabalho dos técnicos responsáveis por monitorizar sistemas complexos e distribuídos. A identificação da raiz do problema e ter a capacidade de decidir qual a solução de forma rápida é essencial para que o sistema não entre em colapso. O desenvolvimento atual contribui de forma significativa para a perceção e rápida análise do sistema, que têm um papel essencial nas ações do utilizador. O próximo passo será a criação de novos gráficos para análise individual ou conjunta de métricas selecionadas nos gráficos de estrutura apresentados. É esperado que o sistema tenha a capacidade de sugerir os gráficos mais indicados mediante os componentes selecionados, bem como a disponibilização dinâmica de *dashboards* com gráficos e organização apropriada a esses componentes.

IX. ACKNOWLEDGMENTS

The research leading to these results has received funding from the [European Union] [European Atomic Energy Community] Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) under grant agreement n° 619606.

Referências Bibliográficas

- [1] J. James, “Domsphere,” 13 Agosto 2015. [Online]. Available: <https://www.domo.com/blog/2015/08/data-never-sleeps-3-0/>.
- [2] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil e D. Barton, “Big data,” *The management revolution. Harvard Bus Rev*, vol. 90, pp. 61-67, 2012.
- [3] M. Zuckerberg, “Facebook,” Junho 2015. [Online]. Available: <https://www.facebook.com/zuck/posts/10102213601037571>. [Acedido em 28 Outubro 2015].
- [4] M. Chen, S. Mao e Y. Liu, “Big data: A survey,” *Mobile Networks and Applications*, vol. 19, pp. 171-209, 2014.
- [5] S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G. L. Rubini, G. Tortone e M. C. Vistoli, “GridICE: a monitoring service for Grid systems,” *Future Generation Computer Systems*, vol. 21, pp. 559-571, 2005.
- [6] I. MongoDB, “Monitoring for MongoDB,” [Online]. Available: <https://docs.mongodb.org/manual/administration/monitoring/>. [Acedido em 28 Outubro 2015].
- [7] T. Xu, “Grid monitoring system survey,” 2011.
- [8] S. Sreeja e S. Chaudhari, “Study on Grid Resource Monitoring and Prediction,” *Procedia Computer Science*, vol. 45, pp. 815-822, 2015.
- [9] S. F. Silva e T. Catarci, “Visualization of linear time-oriented data: a survey,” em *Web Information Systems Engineering, 2000. Proceedings of the First International Conference on*, 2000, pp. 310-319.
- [10] Ganglia, “Ganglia Monitoring Systems,” Janeiro 2001. [Online]. Available: <http://ganglia.sourceforge.net/>. [Acedido em 13 Janeiro 2016].
- [11] M. L. Massie, B. N. Chun e D. E. Culler, “The ganglia distributed monitoring system: design, implementation, and experience,” *Parallel Computing*, vol. 30, pp. 817-840, 2004.
- [12] M. Massie, B. Li, B. Nicholes e V. Vuksan, *Monitoring with Ganglia*, O'Reilly Media, Inc., 2012.
- [13] “Observium Docs,” Observium, [Online]. Available: <http://www.observium.org/docs/>. [Acedido em 13 Janeiro 2016].
- [14] C. Blake, “Observium, the Do-it-All Monitoring Application,” Abril 2013. [Online]. Available: <http://servernetworktech.com/2013/04/observium-the-do-it-all-monitoring-application/>. [Acedido em 13 Janeiro 2016].
- [15] “Nagios,” Nagios Solutions, [Online]. Available: <https://www.nagios.org/about/>. [Acedido em 13 Janeiro 2016].
- [16] C. Joseph, “Open Source Server Monitoring Tools – Cacti, Zabbix, Nagios,” 8 Agosto 2015. [Online]. Available: <https://cloudstats.me/2015/08/08/open-source-server-monitoring-tools-cacti-zabbix-nagios/>. [Acedido em 13 Janeiro 2016].
- [17] E. Simmonds e J. Harrington, “SCF/FEF Evaluation of Nagios and Zabbix Monitoring Systems,” 7 Julho 2009.
- [18] “Wikipedia,” Wikipedia, Novembro 2015. [Online]. Available: https://en.wikipedia.org/wiki/Wikipedia:Graphs_and_charts. [Acedido em 13 Janeiro 2016].
- [19] N. Iliinsky e J. Steele, *Designing Data Visualizations: Representing Informational Relationships*, O'Reilly Media, Inc., 2011.
- [20] A. Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola Jr e R. M. Simpson, “Immersive VR for scientific visualization: A progress report,” *Computer Graphics and Applications, IEEE*, vol. 20, pp. 26-52, 2000.

- [21] E. R. Tufte e P. Graves-Morris, The visual display of quantitative information, Graphics press Cheshire, CT, 1983.
- [22] C. G. Healey, "Choosing effective colours for data visualization," em *Visualization'96. Proceedings.*, 1996, pp. 263-270.
- [23] A. Buja, J. A. McDonald, J. Michalak e W. Stuetzle, "Interactive data visualization using focusing and linking," em *Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on*, 1991, pp. 156-163.
- [24] J. D. Mackinlay, S. K. Card e G. G. Robertson, "Rapid controlled movement through a virtual 3D workspace," em *ACM SIGGRAPH Computer Graphics*, 1990, pp. 171-176.
- [25] S. Engle e S. Whalen, "Visualizing distributed memory computations with hive plots," em *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, 2012, pp. 56-63.
- [26] "D3 Data-Driven Documents," [Online]. Available: <http://d3js.org/>. [Acedido em 04 Fevereiro 2016].